

Copyright © 2004, by the author(s).
All rights reserved.

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

**SPATIAL CHANNEL CHARACTERIZATION
FOR COGNITIVE RADIOS**

by

Jing Yang

Memorandum No. UCB/ERL M05/8

12 December 2004

**SPATIAL CHANNEL CHARACTERIZATION
FOR COGNITIVE RADIOS**

by

Jing Yang

Memorandum No. UCB/ERL M05/8

12 December 2004

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Spatial Channel Characterization for Cognitive Radios

by Jing Yang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

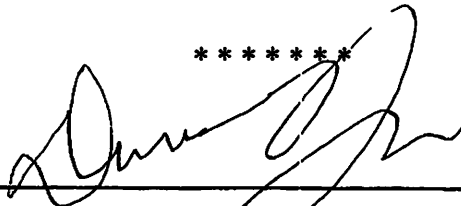
Committee:



Professor Robert W. Brodersen
Research Advisor

12/13/07

(Date)



Professor David Tse
Second Reader

12/14/07

(Date)

TABLE OF CONTENTS

Chapter 1	Introduction.....	1
1.1	Motivation	1
1.2	Scope of This Work	2
1.2.1	Multi-Antenna System.....	2
1.2.2	Summary of Measurement and Processing Work	3
1.3	Channel Model	4
1.4	Organization of the thesis	8
Chapter 2	System Measurement Setup	9
2.1	Transmitter Setup.....	9
2.1.1	Wideband antenna, far field region & antenna array.....	10
2.1.2	Pulse Generator and UWB Signal	16
2.2	Receiver Setup.....	18
2.2.1	Amplifiers.....	18
2.2.2	Data sampler and recorder	20
2.2.3	Antenna positioner	22
2.3	Noise Analysis	23
2.3.1	Introduction to Thermal noise and Input Referred Noise.....	23
2.3.2	Thermal Noise of oscilloscope and amplifiers.....	26
2.3.3	Quantization Noise of oscilloscope.....	28
Chapter 3	Spatial Channel Measurement and Characterization.....	32
3.1	Interference subtraction	32
3.1.1	Approaches to Mitigate the Interference.....	32
3.1.2	Modeling of the Interference and Signal Estimation Problem	36
3.2	Radiation Pattern Analysis	37
3.2.1	Radiation pattern and Reciprocity of the TX/RX patterns	38

3.2.2 Single Antenna Response and Radiation Pattern	41
3.2.3 RX side and TX side radiation pattern	45
3.2.3.1 RX side radiation pattern	46
3.2.3.2 TX side radiation pattern.....	52
3.2.4 Receiver Sensitivity.....	54
Chapter 4 Interference Effect and Cancellation	57
4.1 Introduction	57
4.1.1 Power Spectrum Density Estimation.....	57
4.1.2 Deconvolution Method	64
4.2 Interference at Different Positions.....	71
4.3 Interference Measured from Different Wideband Antennas.....	74
4.4 Spectrum Occupancy and Current Spectrum Overview.....	77
4.5 Proposed Cancellation Method	79
4.5.1 System Concept.....	79
4.5.2 Signal Protection	81
4.5.3 Up convert/Down Convert.....	82
Chapter 5 Future Work and Conclusions	84
Appendix A: Program for Receiver Side.....	86
Appendix B: Program for Transmitter Side	107
Appendix C: Timing Diagram of RX & TX.....	111
Appendix D: Accessing Data Functions.....	112
References	117

LIST OF FIGURES

Figure 1.1 Multipath propagating channel.....	5
Figure 2.1 LLNL antenna shape looking from side-back and top-down....	11
Figure 2.2 Observation sphere	12
Figure 2.3 Transmitting and receiving antenna system	12
Figure 2.4 Two antenna system with conjugate loads	13
Figure 2.5 Two final versions of three antenna arrays	15
Figure 2.6 Pulse generator output (100ps rise time and 35V peak voltage)	16
Figure 2.7 (a)Received waveforms at different receiver azimuth angles... 17 (b) Detail compare when plotted in the same graph	17
Figure 2.8 Received signal with/without the amplifiers.....	19
Figure 2.9 Amplifier board and the SMA cables used to connect oscilloscope.	20
Figure 2.10 Spectrum of received signal with amplifiers.....	20
Figure 2.11 Agilent 54855A oscilloscope.....	22
Figure 2.12 Measurement setup and main components	23
Figure 2.13 Thermal noise of a resistor	24
Figure 2.14 a) Simple transistor amplifier ac schematic.....25 b) Small-signal equivalent circuit with noise sources.....	25
Figure 2.15 Representation of circuit noise performance by an equivalent input noise voltage.	26
Figure 2.16 Diagram for noise calculation	27
Figure 2.17 Trend of the input referred noise decreasing and leveling off as the number of amplifiers goes up	28
Figure 2.18 Ideal ADC input range for N=3bits	29
Figure 2.19 Ideal ADC quantization error for N=3bits	29
Figure 2.20 PDF of a typical quantization error	30

Figure 2.21 Offset error and gain error of ADC.....	31
Figure 2.22 DNL and INL of sample ADC.....	31
Figure 3.1 a) Received waveform when TX angle = 0, RX angle= 0, and average number of the oscilloscope is set to be 4. b) Assume received signal is mainly LOS, Zoom in version shows the detail look of the interference at that time. c) No signal sending out, the pure interference at another time is almost within the same magnitude range, but the time domain waveform is quite different from the last moment.....	33
Figure 3.2 a) Same setup as Fig 3.1, but the average number is changed to 256. b) Same setup as Fig 3.1, but the average number is changed to 1024.....	34
Figure 3.3 Frequency response for different average number points.....	35
Figure 3.4 Radiation pattern and characteristics for a Horn antenna.....	39
Figure 3.5 Antenna rotating and illustration of reciprocity	40
Figure 3.6 System response for some typical receiver angles when TX angle=0	41
Figure 3.7 a) Measured Time vs Azimuth Angle plot for single antenna..	41
b) 3-D Plot of measured signal vs Azimuth Angle for single antenna	41
Figure 3.8 a) Corrected Time vs Azimuth Angle plot for single antenna...	42
b) 3-D Plot of corrected signal vs Azimuth Angle for single antenna	43
Figure 3.9 a) TX antenna response b) RX antenna response.....	45
Figure 3.10 Radiation pattern at different time for single antenna	45
Figure 3.11 System response at different angles for short array	46
Figure 3.12 a) Measured Time vs Azimuth Angle plot for short array	46
b) Radiation pattern at different times for short array	46
Figure 3.13 System response at different angle for long array	47
Figure 3.14 a) Measured Time vs Azimuth Angle plot for long array	48
b) Radiation pattern at different time for long array	48
Figure 3.15 a) After-Correction Time vs Azimuth Angle plot for linear	

array	b) Radiation patterns at different time for linear array	49
Figure 3.16	a) After-Correction Time vs Azimuth Angle plot for cross array	
	b) Radiation patterns at different time for cross array	50
Figure 3.17	Rectangular array with 4 antennas and tongue at bottom.....	50
Figure 3.18	a) After-correction time vs Azimuth angle plot for rectangular	
	array b) Radiation patterns at different times for rectangular array	51
Figure 3.19	a) Linear array b) Cross array c) rectangular array	51
Figure 3.20	Transmitter radiation pattern for rotating 30 degree each step	52
Figure 3.21	Transmitter radiation pattern for rotating 20 degree each step	53
Figure 3.22	Added waveform when TX facing 0°, RX rotating 30° each	
	step	53
Figure 3.23	Time vs Azimuth angle plot for TX	53
Figure 3.24	Nomograph of SNR ratio as a function of probability	56
Figure 4.1	Periodogram plot for Rectangular, Hamming and Kaiser Window	
	61
Figure 4.2	Periodogram and Welch plot for Rectangular window.....	62
Figure 4.3	High/Low SNR: Periodogram plot for Kaiser window	63
Figure 4.4	Flowchart for subtractive deconvolution method.....	67
Figure 4.5	Measurement parameters for LOS, ceiling and ground bounce	68
Figure 4.6	Ground bounce (top), LOS (middle), Ceiling bounce (bottom)	69
Figure 4.7	Flowchart for Multi-template subtractive deconvolution method	
	70
Figure 4.8	Dirty Map of interference at position in meeting room.....	72
Figure 4.9	Dirty Map of interference at position next to wireless LAN	
	transmitter	73
Figure 4.10	Dirty Map of interference at position in the Lab next to BEE	73
Figure 4.11	Dirty map ranged from 0 to 6GHz with a)LLNL b)TEM Horn	
	74
Figure 4.12	Equipment Noise between 0~6GHz with a)LLNL b)TEM Horn	
	76

Figure 4.13 Interference map between 0~6GHz with a)LLNL b)TEM Horn 76

Figure 4.14 Current Frequency allocations in United States from FCC 77

Figure 4.15 Concept of the interference cancellation system..... 80

Figure 4.16 Multi-user Detection 82

Figure 4.17 Up convert to do cancellation..... 83

Figure 4.18 Down convert signal to do cancellation..... 83

Figure 4.19 Distortion causes unwanted interference in signal band..... 83

LIST OF TABLES

Table 2-1 Far field region measurement data.....	14
Table 2-2 Input referred noise versus No. of amplifiers.....	28
Table 3-1 Tolerable measurement time.....	54
Table 4-1 Usage percentage of spectrum taking at Berkeley Downtown ..	78
Table 4-2 Position, BW and Magnitude of the existing signals at 3~5GHz range.....	79

Acknowledgements

There is no way for coming through this project and bringing my first outcome without the help from these people and institutions. First of all, I would like to thank my advisor Professor Bob Brodersen for his great support and guidance, even stopping during his busy day to help in a debug error with me when doing the measurements. His insight and broad knowledge helped me avoid going down wrong path several times and shortened the path to achieve the project goal. His personality and marvelous reputation in different areas is a role model for all his students from both the living life side and the academic side. I would also like express my thanks to Professor David Tse, whose lectures gave me a solid background and strong intuition in the communication field, which was used in part of this work in a deeper exploration on the observed data. Great thanks go to Dr. Ada Poon, as a mentor of a grad student diving into a total new area as me, she was patient and very thoughtful in giving hints and defining the project. Her diligent and active attitudes of living life make a significant impression on me.

Next, I would like to thank Ernest Tsui, who carefully proofreading this thesis and gave a lot of encouragement when I was in bad mode and Gordon Wan, who is the main support of writing the program. Thanks also to my great colleagues at BWRC: Stanley Wang, Cabric Danijela and Mike Chen all helped me a lot in aspects such as setting up the equipment, analyzing the data and giving ideas of the possible focus to the project. Nathan Pletch, Ruth Wang, who are my same year dear friends, and I boat together through the darkest days of preparing for the preliminary exam. Brian Otis, Changchun Shi, Ian O'Donnell, Sohran Emami-Neyestanak, Dejan Markovic and Shridhar Mubaraq Mishra also kindly provide me very helpful information in doing the measurements.

Last, but not least, my parents and my boyfriend, Wei Zheng. Although life is hard, but you are always the persons who support me and take care of me before this thesis is done. Without these endless and priceless love, it will never be possible for me to still live happily and accomplish my tasks. I feel great fortune to have you all accompany with me, only in this way, my life is complete and honorable. Finally, my friends in China and in Berkeley, I will not forget the time we spent together before and in the future. Thank you all!

Chapter1 Introduction

1.1 Motivation

The regulation of spectrum has had almost 100 years to become one of the most political of bureaucracies. The interests of some of the most powerful companies in the world have had much influence over the direction and focus of the Federal Communications Commission (FCC). The FCC micromanaged spectrum allocations on a request-by-request basis, usually specifying the application (TV broadcast, phone service, public safety, etc.) and technology that a licensee could use its sliver of spectrum.

Until recently, there was little expectation for this to change, however the past few years or so, the FCC itself realized that there is a need for change. It basically had no more spectrum to allocate yet the demand for new uses—primarily data—was accelerating. Fortunately, the FCC was open to new techniques for using spectrum and at the same time discovered that much of the spectrum allocated over the years was being grossly underused. The problem is that the new techniques to take care of underused spectrum and bringing it under control require fundamental new ways of regulating it.

Smartly sensing and using “underused” bands, which includes both the “unlicensed” and “licensed” bands, will avoid the impending crisis of spectrum availability at frequencies which can potentially be used for wireless communications.

Three major technologies proposed to implement spectrum sharing are:

- Low-power, wideband spread spectrum underlay
- Cognitive/agile/software-defined radios
- Mesh networks

In addition, the advantage of antenna array systems for both interference cancellation and creating better signal beamforming applications requires the need to better understand the spatial properties of the wireless communications channel. These spatial properties of the channel will have an enormous impact on the performance of systems; hence, an understanding of these properties is paramount to effective system design and evaluation.

With our focus on cognitive radio, therefore, the goal of this project is to create a basic understanding of what the channel looks like over a wide frequency bandwidth for both signal and interference and how the spatial diversities can be improved by using the antenna array. A system level specification for our cognitive radio design will also be presented.

1.2 Scope of This Work

One objective of this work is to measure the channel response by sending an Ultra-Wideband signal and also analyze the interference effects in the frequency domain. The measurement will enable new ways to cancel the interference while improving the received signal quality. By manipulating the antenna array setup and minimizing input referred noise, we can locate interference very precisely and achieve a relatively clean signal.

1.2.1 Multi-Antenna System

The Multi-antenna system consists of two or more antenna elements that are

spatially arranged and electrically interconnected to produce a directional radiation pattern. The interconnection between these elements can provide more information on amplitude, phase, etc., which helps to improve the capacity of a wireless communication system.

1.2.2 Summary of Measurement and Processing Work

In this measurement, a 100ps rise time pulse generator is used to generate a sharp rising edge wideband signal; one single 100MHz ~ 6GHz wideband antenna works as a transmitter, sending the signal in a relatively empty space free of obstacles to get a line of sight (LOS). At the receiver side, we setup an antenna array, with different configurations, such as linear, rectangular and cross shapes. Considering the real channel, there are two major factors that will affect the measured data, the thermal noise from all our equipment and the interference generated from the primary user throughout the band from 0 to 6GHz.

To get a good estimation at the receiver side, it is important to do an investigation on how to minimize the interference and the input referred noise. Without the compensation, the noise will be too large and may even dominate the output. Here we need to emphasize one of the most important assumptions in all our remaining work and design, which is, the thermal noise needs to be small and will not twist the received signal far away from where it should be. Therefore, with good noise immunity, a second step in getting the radiation pattern for different array configurations is explored. Two main purposes of measurements are done, azimuth only and both azimuth and elevation.

Since we are cognitive radio oriented, being a secondary user (SU), learning how the primary user (PU) is working will be the first lesson that we are going

to take. But, obviously, when the SU is working, the PU will act as interference. By determine a way to get the most reasonable power spectrum density (PSD) and doing the deconvolution on the basis of known PSD, a clean map of the current interference can be obtained in the 0 to 6GHz band. Interference locations as well as magnitudes allow us to specify such system parameters, such as dynamic range (DR) and sensitivity.

In this thesis I proposed an effective way to do interference cancellation, which, by adding some transmit constraints to the transmitter side, is the core factor for determining whether our cognitive radio may or may not perform adequately and will greatly improve the performance of our cognitive radio as a secondary user.

1.3 Channel Model

In a wireless system, a signal transmitted into the channel interacts with the environment in a very complex way. There are reflections from large objects, diffraction of the electromagnetic waves around objects, and signal scattering. Another common concern of wireless channels is the presence of Doppler shift, which is caused by the motion of the receiver, the transmitter, and/or another objects affecting the channel. But, since our system mainly deals with high symbol rates, and slow movement applications, the Doppler shift will be small and negligible. So in our channel model, we will not include this time varying factor of Doppler shift. The final result of all these complex interactions is the presence of many signal components, or multipath signals, at the receiver. Each signal component experiences a different path environment, which will determine the amplitude/phase, time delay, and angle of arrival, and etc. In general, each of these signal parameters will be time varying. In this work, the more accurate model is going to be used, involving in the concept of time delay

spread, representing the RF channel as a time-variant channel and using a baseband complex envelope representation, the channel impulse response of such a channel is represented as

$$h(t) = \sum_I A_i(t) \delta(t - \tau_i(t))$$

Where I is the number of multipath components, $A_i(t)$ is the amplitude of the multipath components.

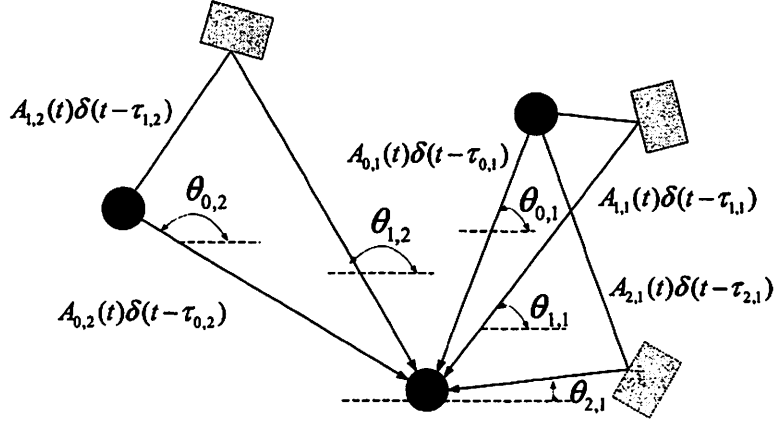


Figure 1.1 Multipath propagating channel

If we want to make our model more accurate, thus more complex, the parameter of angle of arrival can be included and $\vec{a}(\theta(t))$ is what people usually called array response vector, which is a function of both the array geometry and the angle of arrival.

$$h(t) = \sum_I A_i(t) \vec{a}(\theta(t)) \delta(t - \tau_i(t))$$

For an arbitrary array (with a fixed geometry shape), say, limited for two-dimensional space, the resulting array response vector is given by

$$a(\theta(t)) = \begin{bmatrix} \exp(-j\psi_1) \\ \exp(-j\psi_2) \\ \exp(-j\psi_3) \\ \dots \\ \exp(-j\psi_m) \end{bmatrix}$$

where $\psi_i(t) = [x_i \cos(\theta(t)) + y_i \sin(\theta(t))] \beta$ and $\beta = 2\pi / \lambda$ is the wave number.

The spatial channel impulse response given above is a summation of several multipath components, each of which has its own amplitude, phase and angle of arrival. The distribution of these parameters is dependent on the type of environment. In particular, the angle spread of the channel is known to be a function of both the environment and the base station antenna heights. And all these are based on that in our channel model, we want to resolve the impulse response components into bins smaller than the duration of the sounding pulse and smaller than the reciprocal of the channel bandwidth. Considering this desire, that we want to characterize the channel for different TX and RX angles and we will add these two parameters into the model. Assuming no time variation and just considering a fixed array shape for a moment, now the channel model we are using can be represented as a vector or matrix, with each component described as below:

$$h(t, \theta_t, \theta_r) = \sum_i^I A_i(t, \theta_t, \theta_r) \hat{h}^i(t - \tau_i)$$

Three important things to note are the following:

First, we avoid using the $\delta(t - \tau_i)$ function. Because the channel model we described before is adequate for narrow band communication purposes, while for wideband and even UWB, the delta function at the receiver implies an infinite channel bandwidth, which is not possible or realizable. To make the model more accurate, the reference pulse used is the convolution of the sounding pulse, transmitter and receiver antennas, and the impulse response of the detector (sampling oscilloscope, Figure 2.11 Agilent 54855A oscilloscope).

Practically, the reference pulse is measured in a well-behaved channel where the multipath reflections can be gated out. The transmitter and receiver are placed face to face with a relative large distance to guarantee the far field reception for the given antenna that is considered. Later we will elaborate more

on our antenna, the measurement and how we guarantee the far field region.

Second, we denote $\hat{h}^j(t - \tau_j)$ as the j th template of system impulse response when excited by the line-of-sight pulse. When the received signal is the line-of-sight then it corresponds to $\hat{h}^1(t) = \delta(t)$. There will be quite a number of templates, based on the observation that the multipath components have different shapes to that of the reference line-of-sight template and the same conclusion can be extended to all practical antennas. So this channel model allows for more than one received pulse waveform.

Finally, the transmitter and receiver angles give several kinds of combinations and are used to define the matrix. Either transmitter or receiver angle can be two-dimensional or three-dimensional. The total number of measurements, thus the size of the matrix, depends on both the resolution angle (e.g. 5°), and the time we may spend on doing the measurements. It is much easier to deal with a two-dimensional TX or RX angle one by one, but on the other hand, the space and channel are really three dimensional, so the choice will strongly depend on the real need. But we can clearly see that the size of the matrix (which will later be used as a look up table for the future research) increases significantly. When doing three-dimensional measurement, it will be proportional to the set of measurements of TX angle doing in azimuth \times set of measurements of TX angle doing in elevation \times set of measurements of RX angle doing in azimuth \times set of measurements of RX angle doing in elevation, which will become a huge number, and turns out both impossible to do the measurement and to use the existing computer to do the analysis. Our final choice is to make TX rotates only in azimuth domain and RX rotates also only in azimuth angle. To get a quick glance on what is happening on the reflection of roof and floor, several three-dimensional measurements are carried out, with the waveforms plotted and analyzed.

1.4 Organization of the thesis

The remainder of the thesis has four main chapters. Chapter Two builds up the basic system setup for both transmitter and receiver, giving background of what is the special characteristic of wideband and UWB signal and channel measurements. Introduction of thermal noise is presented which will play an important role for our whole measurement, both in terms of setup and providing information on creating noise immunity and doing cancellation. Chapter Three presents an analysis about how to deal with interference and noise problems, in order to guarantee the results are accurate. Also in this chapter, the radiation pattern, beam forming and sensitivity of both the transmitter side and the receiver side for different kinds of array shape will be covered and the received signal will be presented from time, frequency and spatial domain perspectives. In chapter Four, the interference effects are shown. The way to mostly effectively deconvolve the interference in frequency domain is explored, based on this tool, we present measurements in a real environment, downtown Berkeley, to get the current spectrum occupancy. The measured data shows, not surprisingly, the spectrum is far more from being crowded, we give the percentage, position and bandwidth for all the possible interference at a certain time and location as an illustration. Mostly importantly, a proposal for interference cancellation is brought out and may be of interest for future research. Finally, Chapter Five concludes with the status of the current work and future possible tasks.

Chapter2 System Measurement Setup

2.1 Transmitter Setup

Based on the spectrum of interest, which spans from relatively low frequencies, like hundreds of MHz, to several GHz, we need to choose the antenna, which does not degrade the signal for ultra wide bandwidths. In order to operate in far field region, we need to keep the distance between the transmitter and the receiver large enough, and some basic measurements to test if a selected distance is a far field region or not is necessary. When getting the measurement result, we will be able to tell, at least, to keep in far field region, how far should the transmitter and receiver be away.

For the antenna, another important aspect of its characteristic is to have a sense, before doing the measurement and characterization, about what the signal should approximately look like. That is, if sending out a certain kind of signal, how many derivatives are we expecting, what in time domain the signal should look like approximately.

Besides the antenna, we definitely will choose this “certain kind of signal”. How good the signal will be would, to some extent, affect our accuracy and the effort to clean the signal. With the existing equipment and technology, we make our choice to be a pulse generator, which generates a periodic signal with its repeating frequency variable from 10Hz to 100KHz, and its maximum peak is 35V, which can be attenuated by 100dB. Regarding the shape of the pulse, it has a very sharp rising edge with rise time as small as 100ps, can be delayed up to 40ns, followed by a very low rate fading tail, so that we may approximate it

as a step. After one derivative from the transmitter antenna, it becomes an impulse and if we assume the receiver also perform another derivative at the antenna, then we will get a LOS with two derivatives. This assumption will be tested later and possible explanation will be explored.

2.1.1 Wideband antenna, far field region & antenna array

- *Wideband antenna requirement and special characteristics*

For most of the communication system, antennas with 50ohm radiation resistance as characteristic impedance are used. Most of the measurement of UWB is carried out with these kinds of antennas, so we will also. On choosing the antennas, there are a couple of issues. First of all, we want to really achieve Ultra-wideband, so the combination of antenna system and the measurement equipment should give us a wide frequency range optimally 0~10GHz. Next, considering the antenna positioner's holding capability and power of the motor for rotating, very large and thus heavy antennas should be avoided. Finally, due to its application in short-range impulse system, the antenna should be ultra wideband with linear phase characteristic and with constant polarization across frequency (this may not be as important as the first two requirements). These constraints limit the available antennas that could be considered as candidates, since, commonly saying, the antenna that has wide frequency range is to some extent, too large to be tolerant. While, some newly developed antennas [Seong], which possibly meet both our requirements are not yet available, so we need to trade off the size and the frequency range.

LLNL antenna is that certain antenna that can give us a balance point. It has a frequency low pass filter characteristic, and the flat frequency response enables a display of the whole 0~6GHz range signal that interests us, while, at the same

time, these copper made antennas' diameter are around 4inch, which make them easy to maneuver.

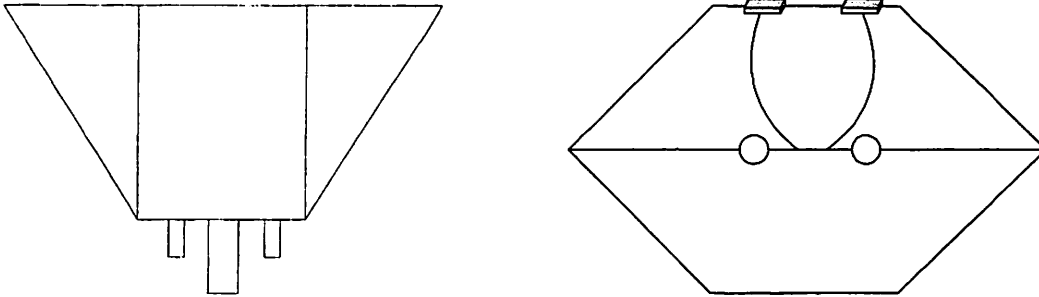


Figure 2.1 LLNL antenna shape looking from side-back and top-down

For these kinds of UWB antennas, the signal sending out and receiving has some very special properties. We will observe that from the transmitter side, the waveforms radiated in different directions from the transmitter antenna will be quite different. And later in the receiver side, again, it will be pretty clear that the impulse response at different positions look considerably different in far field region, too. If we calculate the center frequency of the wavelength and compare it to the size of the antenna,

$$\lambda_{center} = \frac{3 \times 10^8 \text{ m}}{4 \times 10^9 \text{ Hz}} \approx 3 \text{ inch} < S_{antenna}$$

it is clear that the antenna is electrically large compared to the wavelength of the center frequency of the received signal. Under this condition, the above two observations at the transmitter side and the receiver side is reasonable and is generally true.

Looking at the fact that we will rotate the transmitter and the receiver, it will be interesting to do some research on exploring if the reciprocity will still hold under these circumstances. Like in other fields, reciprocity theorem can be applied to electromagnetic theory. To demonstrate its potential, consider an antenna example first. Two antennas, whose input impedances are Z_1 and Z_2 , are

separated by a linear and isotropic medium, as shown in the Figure 2.2, one antenna (#1) is used as a transmitter and the other (#2) is used as a receiver.

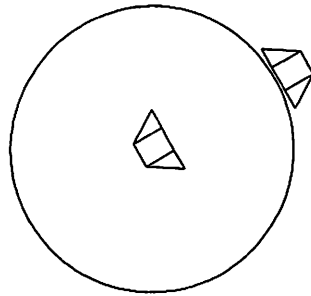


Figure 2.2 Observation sphere

The equivalent network of each antenna is given in Figure 2.3. The internal impedance of the generator Z_g is assumed to be the conjugate of the impedance of the antenna #1 ($Z_g = Z_1^* = R_1 - jX_1$) while the load impedance Z_L is equal to the conjugate of the impedance of antenna #2 ($Z_L = Z_2^* = R_2 - jX_2$). These assumptions are made only for convenience.

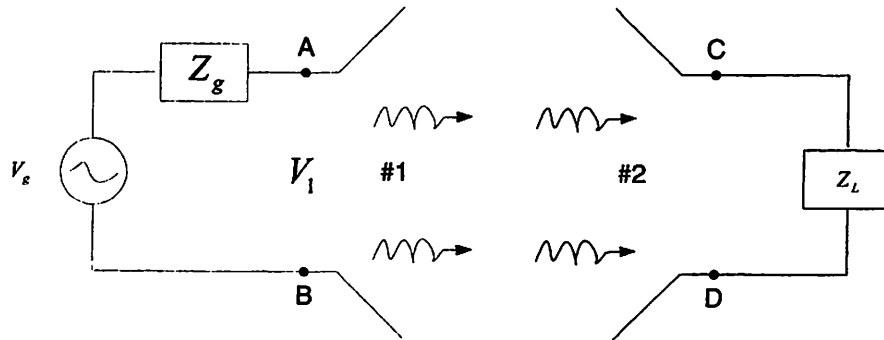


Figure 2.3 Transmitting and receiving antenna system

The power delivered by the generator to antenna #1 is given by

$$P_1 = \frac{1}{2} \text{Re}[V_1 I_1^*] = \frac{1}{2} \text{Re} \left[\frac{V_g Z_1}{Z_1 + Z_g} \frac{V_g^*}{(Z_1 + Z_g)^*} \right] = \frac{|V_g|^2}{8R_1}$$

If the transfer admittance of the combined network consisting of the generator impedance, antennas, and load impedance is Y_{21} , the current through the load is $V_g Y_{21}$ and the power delivered to the load is

$$P_2 = \frac{1}{2} \operatorname{Re}[Z_2 (V_g Y_{21})(V_g Y_{21})^*] = \frac{1}{2} R_2 |V_g|^2 |Y_{21}|^2$$

the ratio of the power becomes

$$\frac{P_2}{P_1} = 4R_1 R_2 |Y_{21}|^2$$

In a similar manner, we can show that when antenna #2 is transmitting and #1 is receiving, the power ratio of P_1/P_2 is given by

$$\frac{P_1}{P_2} = 4R_2 R_1 |Y_{12}|^2$$

under conditions of reciprocity ($Y_{21}=Y_{12}$), the power delivered in either direction is the same.

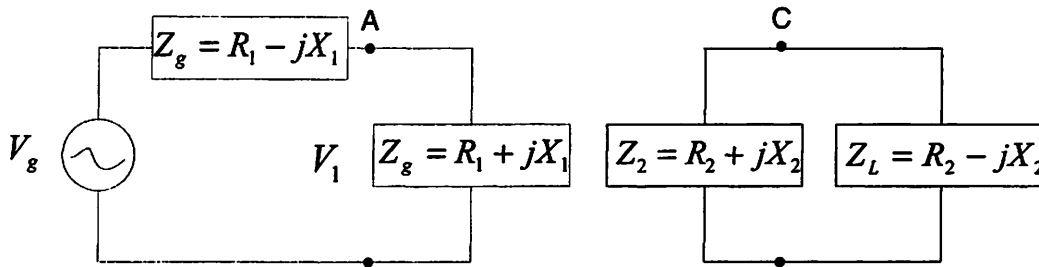


Figure 2.4 Two antenna system with conjugate loads

We will discuss more about the reciprocity for radiation patterns in the next chapter.

- *Far field region testing*

The whole testing being done under far field region is a basic and fundamental requirement and should not be ignored. Although we are not demonstrating what is the exact boundary for the near field region and far field region, but at least we know a lower limit so that we will guarantee that each time we do the measurement, it is far enough to at least make the data reasonable. The far field region testing was done by measuring at discrete distances using the frequency

synthesis and spectrum analyzer. By observing the data as in the following Table 2.1, and knowing that under the far field region condition, the power should be inversely proportional to the distance squared between the transmitter and receiver, we will be able to have a sense of how far to place the antennas.

Distance (inch)	231	198	166	128
(m)	5.87	5.03	4.22	3.25
Signal (mV)	24	28	33	35
Ratio of P_r^2	1	0.9997	0.9885	0.8074

Table 2-1 Far field region measurement data

We will see later when the antenna array system involves in, if we will be able to place the antennas as far as possible, that will be the best solution, because we can, at one side, guarantee the far field region for sure, and at the other side, the received signals from all antennas placed in the antenna array can be added up and averaged directly without any geometric correction. However, our best solution is limited by the spatial room and the transmitting power we will be able to provide. Clearly from the table above, we can see that when distance is larger than 5 meter, the signal power we received almost exactly follows the law (the error is mostly coming from our very noisy measurement), when distance drops to 4.2m, this law is violated a little bit, while distance comes down to 3.2m, it is sure that we already out of the far field region. Finally, we made the choice of placing the TX and RX further than 5m.

- *Antenna array*

On setting up the antenna array, we wanted first to build up a relative flexible/reconfigurable array, which allows us to place the receive antennas in

any combination shape in a plate formed by the array. We will be able to adjust the distance between the antennas to the center of the array, rotate the antenna left/right for 45° (because the tongue is bias to one side, so left and right should be in some sense two difference directions), or place four antennas in one line or two dimensional space, like the cross shape or rectangular.

Instead of a highly accurate, but over-complicated model which may blind the goal by too much focusing on the detail, we wanted a simple and easy description of the channel which may be rough, but enough for understanding and helping the design of the RF system.

Realizing the problems that exist, we simplify our array set up as following: we basically, create three arrays using very light wood boards, with one long linear array where we can place the antennas in a row and the distance between each pair of antennas is larger than the size of the antenna itself; one short arm cross array, which when placed on the positioner lie one arm horizontal and one arm vertical; one short arm rectangular array, which has the same shape as the cross array, but is placed 45 degree between horizontal and vertical for both two arms as shown in Figure 2.5.

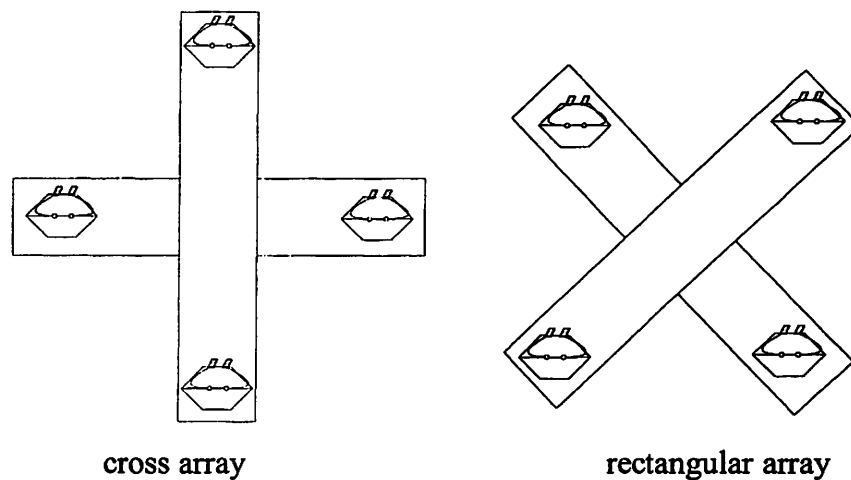


Figure 2.5 Two final versions of three antenna arrays

2.1.2 Pulse Generator and UWB Signal

As discussed previously, to avoid dealing with distorted signals and the required algorithms to correct them, we tried our best to find a clean signal as the stimulus for the whole system. The pulse generator we used is from Picosecond Pulse Labs, which generates periodic signal with sharp rising edge and the peak voltage can be as high as 35V.

The three most important specifications were,

- Shape: Rising time of the edge is as small as 100ps, can be delayed up to 40ns, followed by a very low rate fading tail, so that we may approximate it as a step.
- Amplitude: Peak voltage is 35V, and can be attenuated by 100dB at most, so it enables long distance transmit and strong interference tolerance.
- Repeating frequency: the minimum frequency can be as low as 10Hz and it can be adjusted all the way up to 100KHz, which provides more options when we need to make decisions depending on the measuring time and the specification of the oscilloscope.

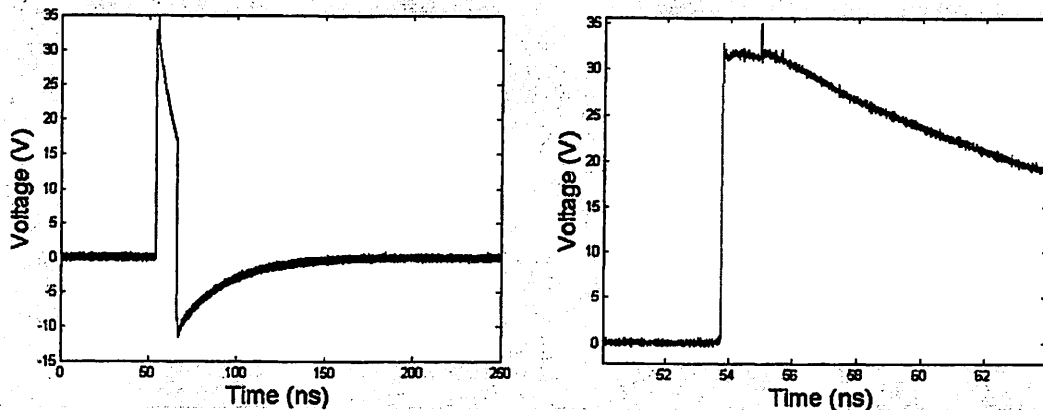


Figure 2.6 Pulse generator output (100ps rise time and 35V peak voltage)

It is interesting that many recent research paper showed that the received pulses

through different paths have the same waveform is not justified for the UWB signal. In [Muqaibel], it is stated that only when both the transmitter and receiver antennas have spherical patterns at all frequencies may come the observation that the waveform received is the same at different paths. However, as we have already considered at the antenna section, our antenna is electrically large compared to the wavelength of the center frequency of received signal, so the assumption should be satisfied, the waveforms radiated in different directions from the transmitter antenna look considerably different in the far field region.

Below, we show that the waveforms received at different angles look different under a simple scenario where the transmitter and the receiver are facing each other and the receiver azimuth angle is rotated 15° in each step.

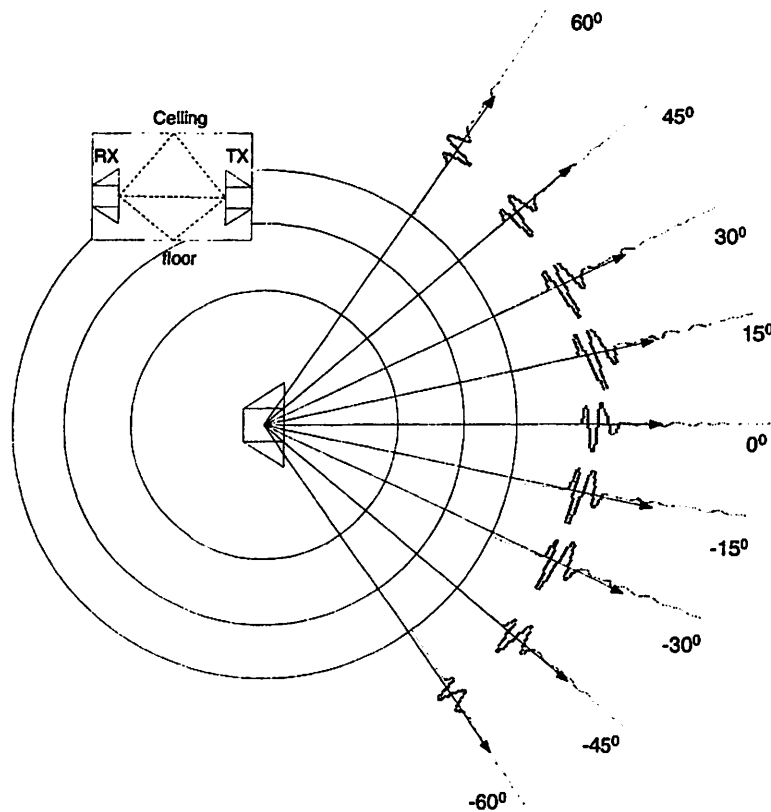


Figure 2.7 (a) Received waveforms at different receiver azimuth angles

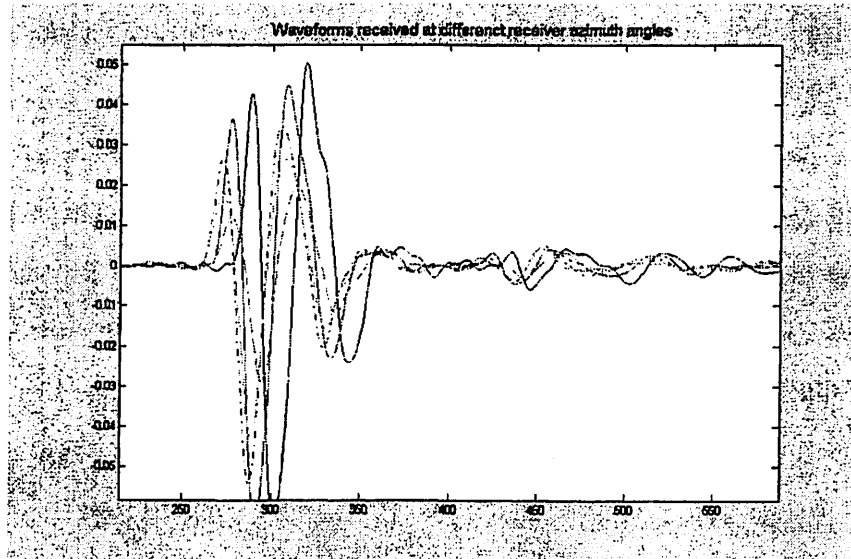


Figure 2.7 (b) Detail compare when plotted in the same graph

2.2 Receiver Setup

If we regarding the channel as fading, time varying channel, which is commonly true for all real channels, it will be very important that the sensitivity at the receiver side is sufficient and the sampling rate we set for the oscilloscope is fast enough so that we can believe the data. Amplifiers are needed to boost the signal magnitude and at the same time not bring in any distortion. The amplifier noise can be dealt with, so that it doesn't decrease the system level noise figure significantly. The oscilloscope not only provides us a very good sampling speed, but also gives us an easy way to record the data.

2.2.1 Amplifiers

When first measuring the data under the far field region condition, we get the maximum received signals in the range about 3 to 4mv, whereas the sensitivity of the scope is 1mV/div, thus it can be seen that we lose a lot of details in terms of accuracy. There comes the problem that how do we boost the signal (of

course the noise level will also rise up) and to what level, so that it does not affect the frequency range that we are interested in and still keep the time domain signal moderate so that it doesn't create distortion to the oscilloscope? At the same time, can we get a flatter response in frequency domain and will the received signal be less affected by the interference? And given an ADC, can we get a sufficiently high sensitivity?

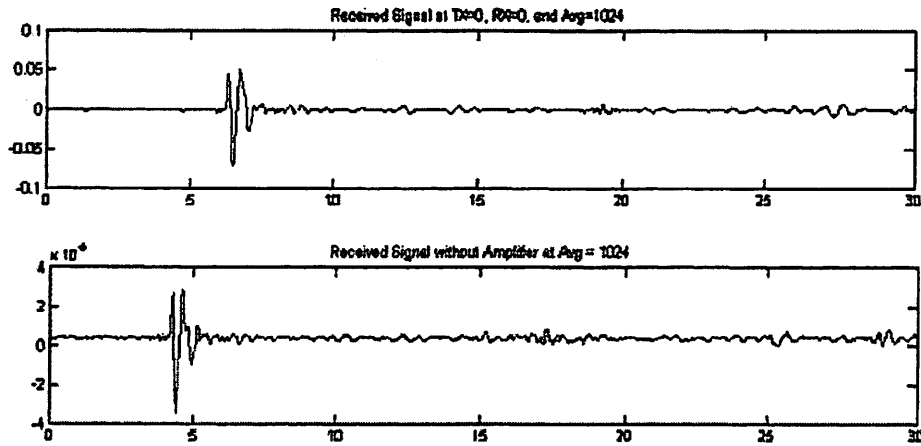


Figure 2.8 Received signal with/without the amplifiers

Another concern is, after making the choice, can we guarantee that enough gain was provided, in terms of avoiding the measurement instrument noise (mainly from the oscilloscope) from dominating the input referred noise?

Finally the amplifier we use is a 12V supply voltage, 4.5dB noise figure, 6GHz bandwidth amplifier with 12dB +/- 1dB gain. We put 8 amplifiers on a board, for each channel, two amplifiers provide approximately 25dB gain and the gain matches very well seeing from the received signal. Figure 2.9 shows the board and the SMA cables that are used to connect the oscilloscope. And the spectrum of a certain received signal with amplifiers is shown in Fig 2.10. We can see now, after the whole setup, the bandwidth of the whole system is 6GHz.

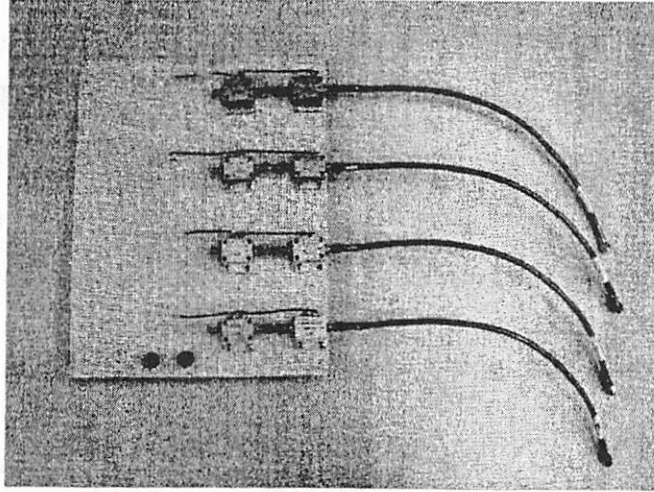


Figure 2.9 amplifier board and the SMA cables used to connect oscilloscope.

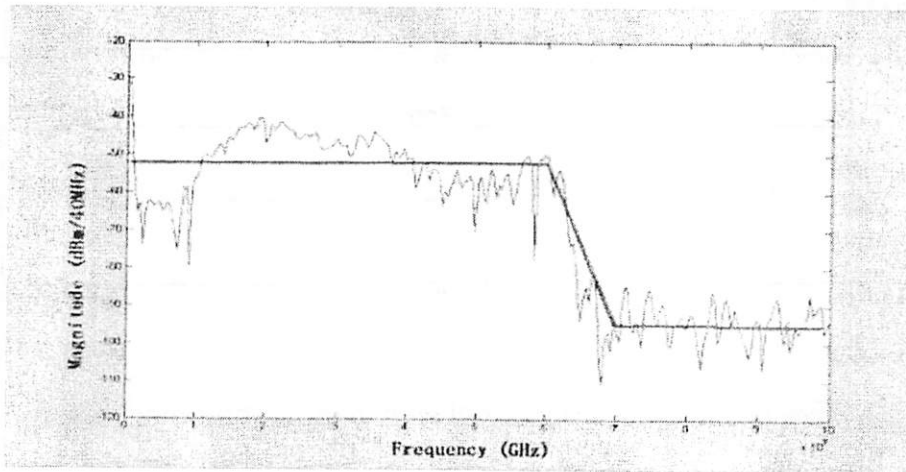


Figure 2.10 Spectrum of received signal with amplifiers

2.2.2 Data sampler and recorder

The Agilent XP integrated-oscilloscope is used in the measurement. The most important characteristic of this scope are that: first, it has four inputs, so this matches the goal that we want to measure the channel with the multiple-antenna system. Given the four inputs, we can avoid bringing extra system error caused by using different equipments. The other one is that it has very fast sampling speed as of 20Gsample/second and can be changed to 10G, 5G sample/second, which brings in some additional flexibility. At the same

time, since it is really a computer so it works very conveniently as a data recording device.

The data was sampled with 20Gsample/sec in our measurement, and then averaged number with a range from 0 (which means no averaging, but just a snap shot of a particular time) all the way to 1024 times, the transmitter and receiver was synchronized externally by using a optical fiber which connects the pulse generator as we mentioned before, with the trigger termination at the oscilloscope.

The reason that the optical fiber is better than the SMA cable is that, under experiments done by AetherWire [Aetherwire] show that when the environment changes, e.g. the temperature goes slightly different on the ground between the transmitter and receiver side or when the ground is shaking a little bit as people walking along, the synchronization of the transmit and receive signal can be violated substantially. As the distance of the transmitter and receiver goes up, the shortcoming of using the SMA cables will be amplified, since the attenuation of the cable worsens and complicates the triggering signal.

After data is sampled, quantized by the ADC inside the oscilloscope, we can directly read the data from the memory of the computer and store it either in binary or decimal number. Another function we used is that we also use the “oscilloscope” to control the antenna positioner accurately by sending out the special command recognizable for the serial port and the telescope.

The program for the Receiver and Transmitter are attached at the end (Appendix A &B), which is used to initialize and control the antenna positioner, read in the data and set the parameter of the oscilloscope, such as sampling rate, averaging number, and offset of each channel.

A picture of this Agilent 54855A oscilloscope is shown in the Fig 2.11. The filter inside the oscilloscope has its 3dB-bandwidth to be 6GHz, which doesn't degrade the total performance of the whole system. The vertical resolution of the ADC inside is 8bit and can go higher than 12bit when averaging.

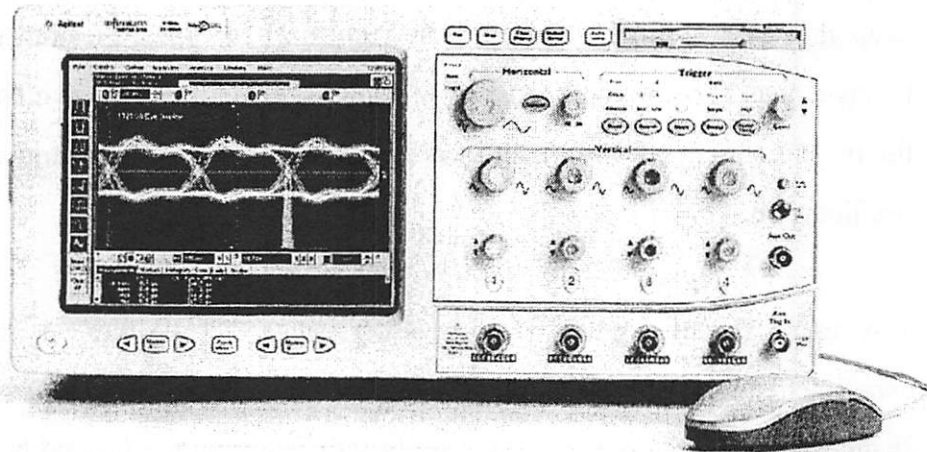


Figure 2.11 Agilent 54855A oscilloscope

2.2.3 Antenna positioner

To hold the antennas and achieve the goal of rotating the array accurately both in azimuth and elevation, one antenna positioner is required for both transmitter side and receiver side. The telescope base motor of Meade ETX90EC and the tripod are used with the special controller called “autostar”, which can accept the command when connected with the serial port of a computer.

Due to the limitation of the “arm”, the maximum degree that the array can go in elevation is from -40° to $+90^{\circ}$, and can go from 0° to 360° degree in azimuth. The motor is powered also by the 12V supply voltage, which is convenient since the amplifiers are powered by 12V. The whole system setup and the main components are shown below.

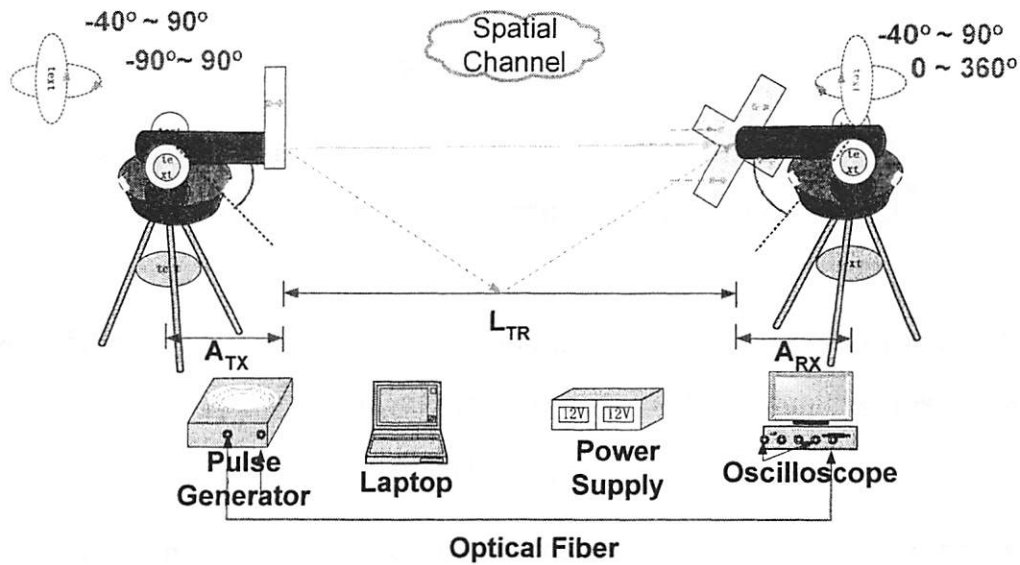


Figure 2.12 Measurement setup and main components

2.3 Noise Analysis

When measuring the signal as well as the interference, the study of noise is important, because it represents a lower limit to the size of electrical signal that can be sensed by a sensor without significant deterioration in signal quality. The two main noise sources we are considering here are thermal noise from the electrical circuit of the oscilloscope and the amplifiers, and the quantization noise of the ADC inside the oscilloscope.

2.3.1 Introduction to Thermal noise and Input Referred Noise

Before getting to input referred noise, let's first look at the resistor thermal noise. The random motion of electrons in a conductor introduces fluctuations in the voltage measured across the conductor even if the average current is zero. Thus, the spectrum of thermal noise is proportional to the absolute temperature.

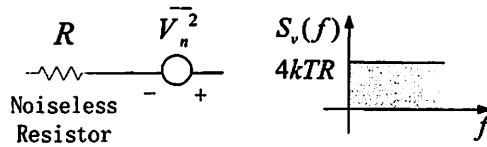


Figure 2.13 Thermal noise of a resistor

As shown in Fig.2.13, the thermal noise of a resistor R can be modeled by a series voltage source, with the one-sided spectral density

$$\overline{V_n^2} = 4kTR$$

where $k = 1.38 \times 10^{-23} \text{ J/K}$ is the Boltzmann constant. For example, a 50ohm resistor held at $T = 300^\circ\text{K}$ exhibits $8.28 \times 10^{-19} \text{ V}^2/\text{Hz}$ of thermal noise. To convert this number to a more familiar voltage quantity, we take the square root, obtaining $0.91 \text{ nV}/\sqrt{\text{Hz}}$. While the square root of hertz may appear strange. It is helpful to remember that $0.91 \text{ nV}/\sqrt{\text{Hz}}$ has little significance per se and simply means that the power in a 1-Hz bandwidth is equal to $(0.91 \times 10^{-9})^2 \text{ V}^2$.

As long as electrical circuits exist, transistors and resistors will bring us thermal noise. Of course, the noise can be calculated at the output, which is so called “output noise”, however, we will see that what is more significant of the noise performance of a circuit is, the limitation it places on the smallest input signals the circuit can handle before the noise degrades the quality of the output signal. For this reason, the noise performance is usually expressed in terms of an equivalent input noise signal, which gives the same output noise as the circuit under consideration.

In this way, the equivalent input noise can be compared directly with incoming signals and the effect of the noise on those signals is easily determined. For this purpose, the circuit of Fig. 2.14 can be represented as shown in Fig. 2.15,

where $\overline{v_{iN}^2}$ is an input noise-voltage generator that produces the same output noise as all of the original noise generators. All other sources of noise in Fig. 2.16 b) are considered removed.

The most natural approach to get the input referred noise is to set the input to zero and calculate the total noise at the output due to various sources of noise in the circuit, we already said that it is not fair comparison of the performance of different circuits, because it depends on the gain. While if we further divide the noise calculated at the output by the signal gain squared A_v^2 , then we can get, the input-referred noise voltage which is what we are looking for. At this point of view, two important observations come out. First, the input-referred noise and the input signal are both multiplied by the gain as they processed by the circuit. Thus, the input referred noise indicates how much the input signal is corrupted by the circuit's noise, i.e., how small an input the circuit can detect with acceptable SNR. For this reason, input referred noise allows a fair comparison of different circuits. Second, the input referred noise is a fictitious quantity in that it cannot be measured at the input of the circuit.

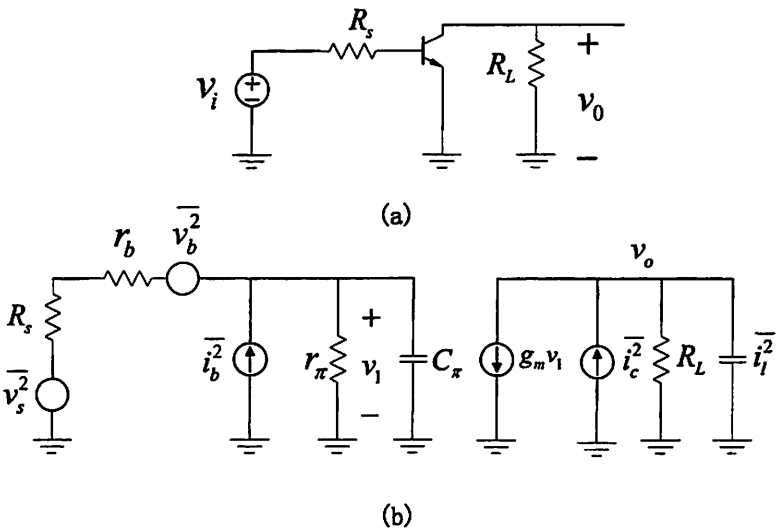


Figure 2.14 a) Simple transistor amplifier ac schematic. b) Small-signal equivalent circuit with noise sources.

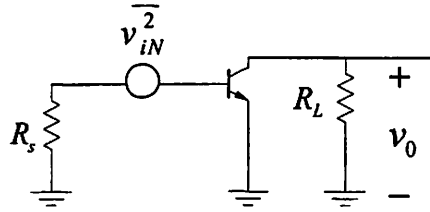


Figure 2.15 Representation of circuit noise performance by an equivalent input noise voltage.

Before we went discussing the detail noise analysis in oscilloscope and amplifiers, let's do a quick and simple introduction to the concept of "noise figure".

Noise figure is a commonly used method of specifying the noise performance of a circuit or a device. It is widely used as a measure of noise performance in communication systems where the source impedance is often resistive. The definition of the noise figure of a circuit is

$$F = \frac{\text{input SNR}}{\text{output SNR}}$$

and F is usually expressed in decibels. The utility of the noise figure concept is apparent from the definition, as it gives a direct measure of the signal-to-noise ratio degradation that is caused by the circuit.

2.3.2 Thermal Noise of oscilloscope and amplifiers

Compared to the quantization noise of the oscilloscope, the thermal noise can really create a disaster. The conclusion was drawn from the measurement of SNDR and the SQNR of the received data from the oscilloscope. In terms of getting the thermal noise of the oscilloscope, a broad band 50ohm resistor is connected to one of the channels as the matching termination and the equivalent noise of the resistor will act as a input to the oscilloscope.

Turning off the averaging option in the oscilloscope, without the amplifier, the signal received and shown on the oscilloscope is 2.5mV, part of which is coming from the broadband resistor, whose noise extends for 6GHz, because the system is indeed a filter and all the input signals above 6GHz are filtered out. We can calculate the equivalent input voltage from the noise source as

$$V_R = 0.91 \times 6 \times 10^9 \text{ nV} = 0.07 \text{ mV}$$

$$V_0 = V_R + V_{scope} = 2.5 \text{ mV}$$

$$\therefore \hat{V}_{scope} = 2.43 \text{ mV}$$

Similarly, from the diagram below in Fig. 2.16, we relate the measured output voltage with the noise coming from the resistor, oscilloscope and amplifiers.

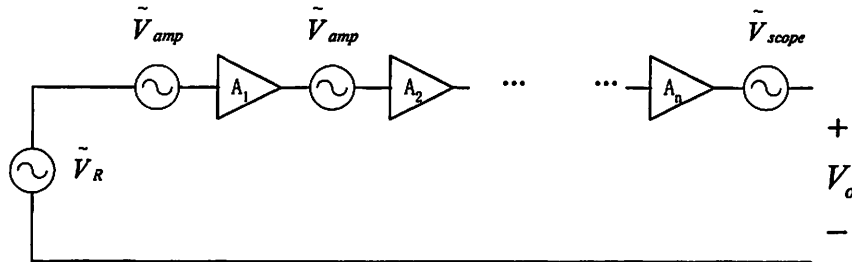


Figure 2.16 Diagram for noise calculation

When $n=1$, which means one amplifier is added,

$$V_0 = (V_R + V_{amp}) \times A_{amp} + V_{scope} = 4 \text{ mV}$$

$$(0.07 \text{ m} + V_{amp}) \times 4.2 = 4 \text{ m} - 2.43 \text{ m}$$

$$V_{amp} = 303.81 \mu\text{V}$$

For sanity check, we calculate the case when two amplifiers are added

$$V_0 = (V_R + V_{amp}) \times A_{amp}^2 + V_{amp} \times A_{amp} + V_{scope} = 10.3 \text{ mV}$$

which is very close to the raw measured output, which is 10mV.

Continue for $n=3, 4, 6$, and calculate the input referred voltage as

$$V_{input-referred} = V_o / A^n$$

Which has been deduced when introducing the concept of the input referred noise, and we get Table. 2.2, plotted and connect the discrete dots, we see the trend of the input referred noise as the number of amplifiers, that is connected before signal reaches the oscilloscope, goes down. It approaches a constant and is dominated by the noise coming from the first amplifier.

Amplifiers No.	Noise p-p (mV)	Gain	Input referred noise
0	2.5	1	2.5 mV
1	4	12.5dB	0.9485 mV
2	10	25dB	0.5623 mV
3	42	37.5dB	0.5601 mV
4	150	50dB	0.4723 mV
6	2500	75dB	0.4446 mV

Table 2-2 Input referred noise versus No. of amplifiers

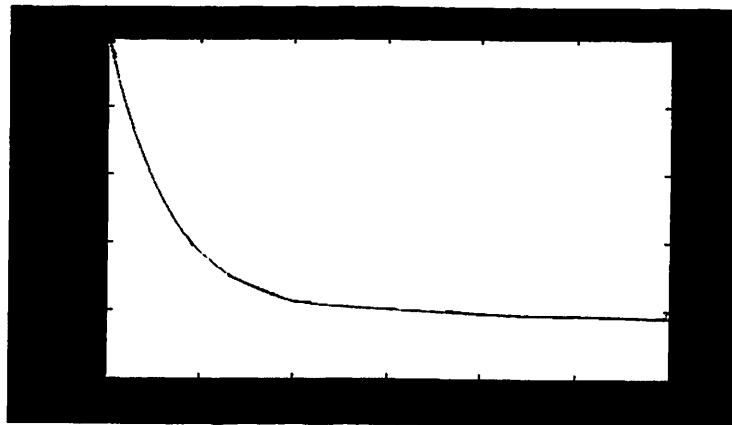


Figure 2.17 Trend of the input referred noise decreasing and leveling off as the number of amplifiers goes up

2.3.3 Quantization Noise of oscilloscope

Quantization noise is inherent in the amplitude quantization process and occurs in the analog-to-digital converter, ADC. The noise is additive and independent

of the signal when the number of levels $L \geq 16$. This is equivalent to $B \geq 4$ bits.

For a given resolution ADC, the dynamic range $DR = 1.76 + 6.02N$ (dB)

e.g. $8\text{bits} \rightarrow 50\text{dB}$
 $12\text{bits} \rightarrow 74\text{dB}$

For an ideal ADC, which can be called the quantizer, we define the quantization step as Δ (LSB). And the full scale input range for the ADC will be $-0.5\Delta \dots (2^N - 0.5)\Delta$, as shown in Fig. 2.18 for an example of $N=3$ bits.

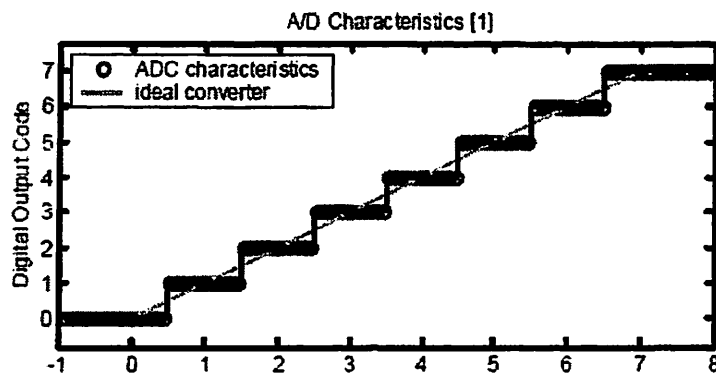


Figure 2.18 Ideal ADC input range for $N=3$ bits

Thus, for input with full scale range, the quantization noise is bounded by $-\Delta/2 \dots +\Delta/2$, as shown in Fig. 2.19 for $N=3$ bits.

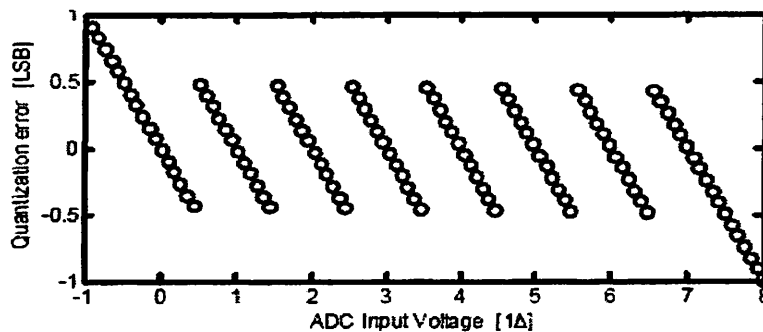


Figure 2.19 Ideal ADC quantization error for $N=3$ bits

If we provided the condition that input is always busy, the number of bits is large enough so that amplitude is many LSBs and finally the ADC is not

overload, then the quantization error PDF can be regarded as uniformly distributed between $-\Delta/2 \dots +\Delta/2$.

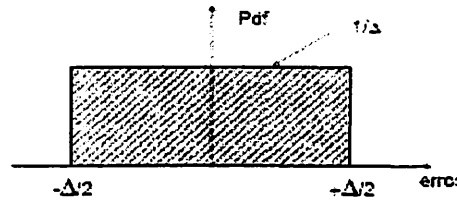


Figure 2.20 PDF of a typical quantization error

We can tell from the PDF that the error will be zero mean, and the variance is

$$\overline{e^2} = \int_{-\Delta/2}^{+\Delta/2} \frac{e^2}{\Delta} de = \frac{\Delta^2}{12}$$

If certain condition is met, the quantization error can be viewed as being “random”, and it often referred to as “noise”. In this case, we can define a peak signal to quantization noise ratio, SQNR, for sinusoidal inputs, and the signal power can be calculated as $\frac{1}{2} \left(\frac{2^N \Delta}{2} \right)^2$, thus the SQNR can be got from the following equation,

$$SQNR = \frac{\frac{1}{2} \left(\frac{2^N \Delta}{2} \right)^2}{\frac{\Delta^2}{12}} = 1.5 \times 2^{2N}$$

real ADC can never achieve sum SNR, because the existence of the thermal noise of the ADC and the deviations from the ideal quantization levels, such as offset, gain error, differential nonlinearity (DNL), and integral nonlinearity (INL). The offset and gain error of the ADC can be, to some extent, compensated by the digital pre or post processor, providing enough measurement and testing. But there is almost no way to avoid DNL and INL.

These two main deviations will degrade the SQNR and can be calculated by just regard it as additional quantization noise due to DNL and INL. Typically, they will degrade the SQNR several dB.

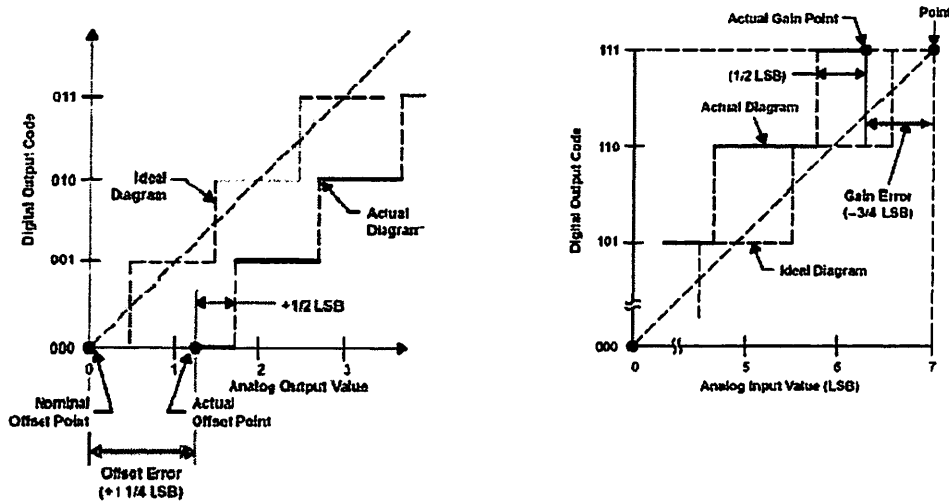


Figure 2.21 Offset error and gain error of ADC

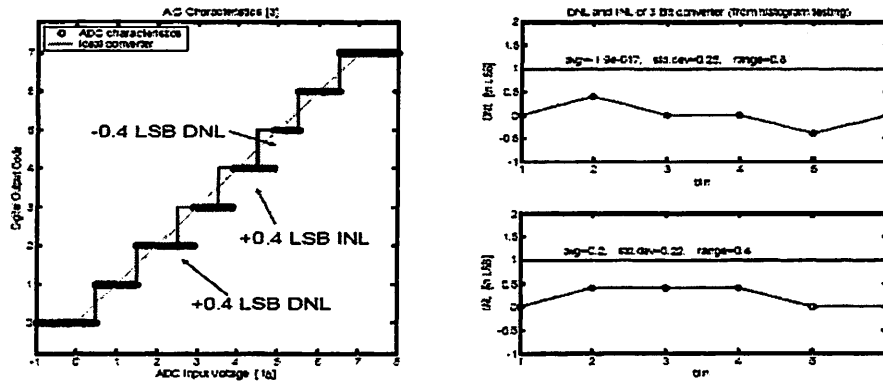


Figure 2.22 DNL and INL of sample ADC

However, if we assume the INL and DNL of the inside ADC is in typical range, when N is 8, as in our system, the actual performance cannot even achieve this high given the consideration of all deviations, since the thermal noise of the amplifier and oscilloscope is far larger than the quantization noise. And in our measurements, the real number of bits for oscilloscope is 8 bits when averaging upto 256 times and 12 bits when averaging 1024 times.

As a summary of this chapter, we described the setup of the whole system providing the reasons and analysis. The noise of the whole system, which will perform a very important role from the sensitivity sense, is dominated by the thermal noise of the first stage amplifier.

Chapter3 Spatial Channel Measurement and Characterization

3.1 Interference subtraction

When analyzing the channel (by sending out the specific signal) we would prefer to suppress the interference effects as much as possible. However, before getting to this point, we would ask the question: is it really a necessity to do the subtraction? If yes, how accurate should the result be and how complex will the algorithm be like? We will provide the approaches and the theoretical modeling to this problem as an explanation.

3.1.1 Approaches to Mitigate the Interference

The filtering solution may come up as the most easiest way, however, because the signal we send out is neither a set of tones, nor narrow band signal in the traditional way, instead, it is wide band signal, so it is difficult to just create a filter and filter the interference at other frequencies; on the other hand, to get high accuracy, the algorithm for getting a perfect subtraction is complicated and hard to implement. So how to trade off the complexity with the accuracy to subtract the interference, getting better performance, while avoiding implementation of a complex algorithm will be the first stage purpose.

Since all the measurements are done in a very empty space, the received signal is mainly LOS and the multipath effect can be ignored. Under this assumption, by first looking at the signals at the receiver side, given the oscilloscope average number to be four, which is very small, we get time domain signal with

interference waveform looks like Fig.3.1 a), zoom in and look at the details at Fig 3.1 b), we can tell the magnitude range of the interference seen on time domain is around $\pm 0.01V$, which is almost one fifth of the peak signal magnitude! Turning of the pulse generator, catching the pure interference at another time, we can see the interference actually changes, so we conclude that the interference varies significantly with time.

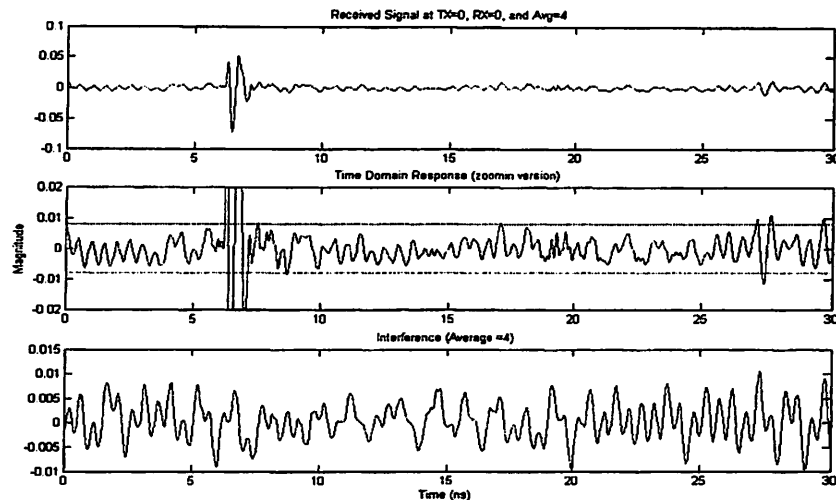


Figure 3.1 a) Received waveform when TX angle = 0, RX angle= 0, and average No. of the oscilloscope = 4. b) Assume RX signal is mainly LOS, Zoom in version shows the detail look of the interference at that time. c) Pure interference (no signal) at another time is within the same magnitude range, but the waveform is quite different from the last moment.

From the waveform we got above, two very important points are noticed here, first, the interference is very large compared to signal itself, mostly because some close by sources of interference are sending out strong signals, which are useful to some clients but are acting as interference to users like us. They are large in magnitude and result in the likely possibility that they might alter the characteristics of the real channel measurement. So, the subtraction of interference must be done and is required for a correct result.

Secondly, given the speed we turn off the signal and sample another set of data,

time constant for taking two sequential measurements is much larger than that of the interference. Thus, the large time variance makes it unreasonable to subtract two sets of data, one of which is recorded when sending signal and the other, just pure interference, shortly after the first set of data. There is no way to guarantee the interference not changing too much during this “short” break.

The way we finally choose to do this first-step simple interference filtering is by manipulating the average number, which is provided as an option in the oscilloscope. We want to reach a point so that we can suppress the interference power as far as we can. Seen from time domain, when average number reaches a certain level, the received signal tail will be much flatter compared to Fig 3.1.

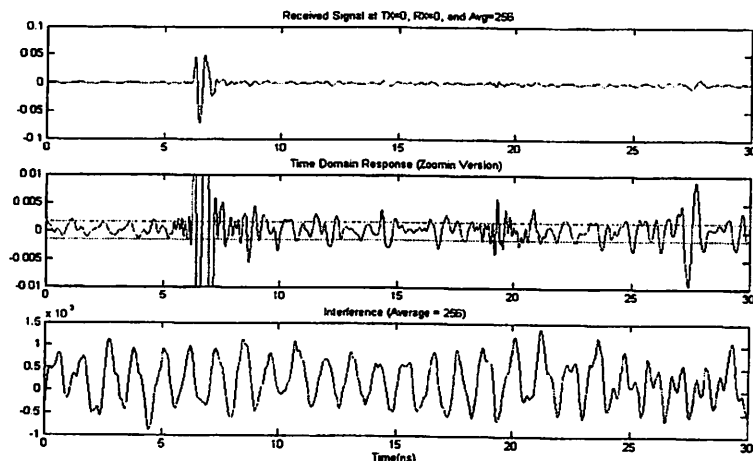


Figure 3.2 a) Same setup as Fig 3.1, average No. is changed to 256

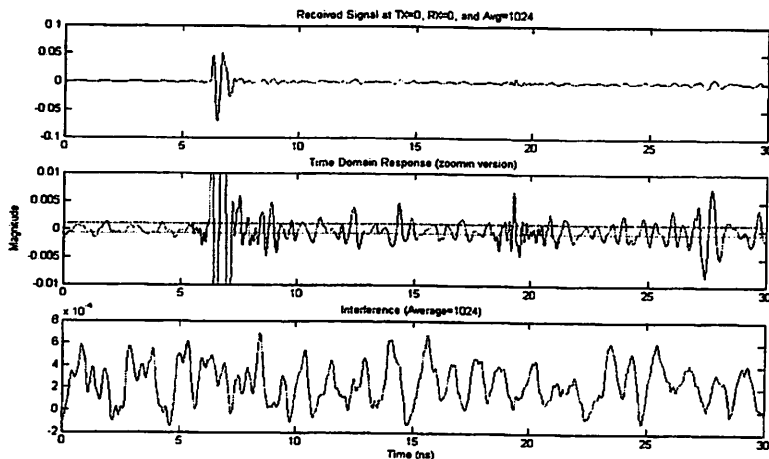


Figure 3.2 b) Same setup as Fig 3.1, average No. is changed to 1024

Now we take a detail look at how the average number changes the frequency response of the whole system. Figure 3.3 shows, without the amplifier, when average number is 4, 256, 1024, the frequency domain plot of the received signal plus interference.

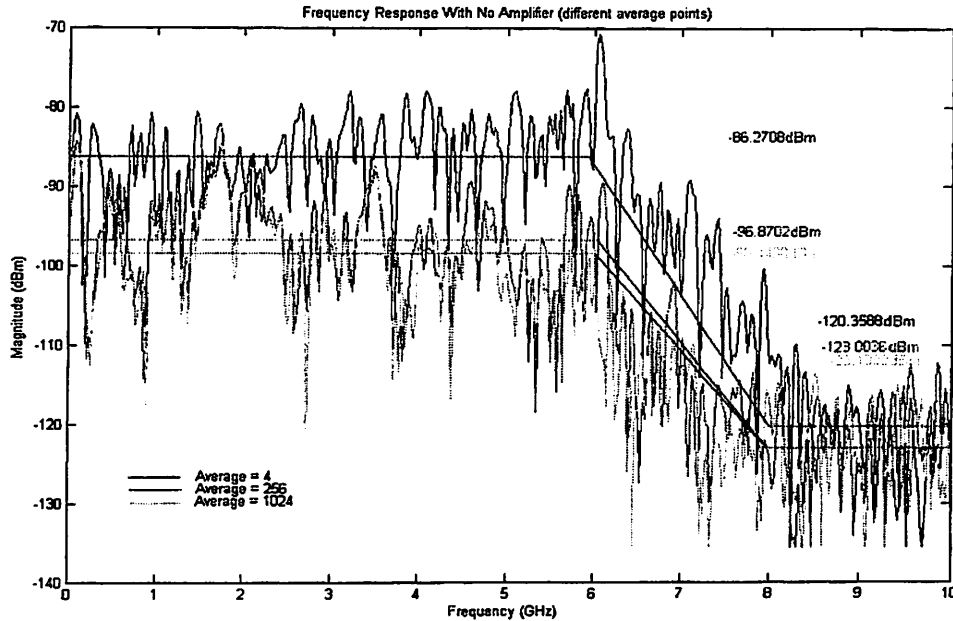


Figure 3.3 Frequency response for different average number points

The averaging reduces the fluctuation of the frequency response. The blue curve shows a lot more fluctuation than the green curve whose fluctuation is relative small although it looks big on the log scale. Its fluctuations are orders of magnitude less, which is consistent with 1024 point average that would give us about 24 dB averaging compared to 4 point average if we can assuming the measurements are independent. The total power level is consisted of both the signal power received and the noise power, which can be divided into white noise and the interference that is colored. From the plot above, since the setup and TX/RX relative position is the same for three curves, the signal power should be equivalent, and the differences between the curves illustrate how much noise was mitigated. By increasing the average number from 4 to 256, power goes down by $(-86.2708) - (-96.6702) \approx 10.4 \text{ dBm}$; increasing to 1024

further, power goes down by $(-96.6702) - (-98.4468) \approx 1.78 \text{dBm}$. Notice that there is big jump for power between using $average = 4$ and $average = 256$, so the interference can be greatly mitigated by changing the average number.

3.1.2 Modeling of the Interference and Signal Estimation Problem

The whole system of using this easy-way interference subtraction can be modeled as below:

We are sending out a certain signal X , and from the oscilloscope we get observations, which can be written as a vector \vec{Y} , $\vec{Y} = \vec{h}X + \vec{Z}$, $\vec{Y}, \vec{h}, \vec{Z} \in R^{n \times 1}$. Since we fixed the TX/RX position and the setup is the same for getting all the observations thus, $\vec{h} = [h_1, h_2, \dots, h_n]^T$, with $h_1 = h_2 = \dots = h_n = h$. Without loss of generality, Let's assume the noise vector \vec{Z} can be modeled as Gaussian noise with zero mean and covariance matrix K_z . In the real system, K_z will have some really nice properties. Because the colored noise is only a very small portion of the whole noise, so most of the Z_i 's are uncorrelated, and under the Gaussian case, independent. Thus let's consider the most roughly true case and assume Z_i 's are IID with covariance σ_z^2 . Then from the parameter estimation point of view, the Maximum Likelihood Estimation (MLE) of X given \vec{Y} can be found by first seen

$$f_{Y_1, \dots, Y_n}(y_1, y_2, \dots, y_n, X) = \frac{1}{(\sqrt{2\pi\sigma_z^2})^n} e^{-\frac{1}{2\sigma_z^2} \sum (y_i - X)^2}$$

and then choose X to maximize the MLE gives

$$\hat{X}(\vec{y}) = \frac{1}{n} \sum_{i=1}^n y_i$$

From the above equation, we can first tell that this estimation is unbiased, since

$E\left[\hat{X}(\vec{y})\right] = X$ and also it is consistent, if as n goes to infinity

$$E\left[\left(\hat{X}(\vec{y}) - X\right)^2\right] = 0$$

Consider the general case now, if Z_i 's are not iid, but have $Cov(Z_i, Z_j) \neq 0$,

$$Var(\hat{X}(\vec{y})) = \frac{1}{n^2} (n\sigma_z^2 + \sum_{i \neq j} cov(Z_i, Z_j))$$

$$\text{if } \frac{1}{n^2} \sum_{i \neq j} cov(Z_i, Z_j) \rightarrow 0$$

then the estimation will be consistent. Assume $\{Z_i\}$ are Wide Sense Stationary (WSS), then

$$cov(Z_i, Z_j) = K_z(j-i)$$

$$\frac{1}{n^2} \sum_{i \neq j} cov(Z_i, Z_j) = \frac{1}{n} \sum_{m=1}^n \frac{(n-m)}{n} K_z(m)$$

so if K_z goes to zero fast enough to make the above equation goes to 0, process will be consistent. This observation is very important since we can intuitively tell that if we want the confidence interval bounded by $[\hat{X} - \epsilon, \hat{X} + \epsilon]$, the longer the data we keep averaging, the smaller the resulting error will be. In another word, if we keep averaging more and more data, the error will be smaller and smaller, and thus the estimation will be more accurate.

3.2 Radiation Pattern Analysis

If we are using an omni-directional antenna to transmit the wanted signal, then space will be circular symmetrical from the TX side point of view, imagining that we have a very sharp beamwidth receiver, which points straightly towards the TX, then we limit our possibility of receiving multiple interferences coming from all directions to a very low level. For instance, suppose the interferences are distributed with equal probability over the entire space, our beamwidth is 10° , then the possibility for receiving an interference is approximately 3%.

From the observation above, we design our system so that we have a pseudo-omni-directional TX and a sharp beamwidth RX. Our purpose is to use this kind of system to avoid receiving strong interference at a certain position, thus boost up the performance matrix of transceiver for a given hardware limit.

3.2.1 Radiation pattern and Reciprocity of the TX/RX patterns

Radiation pattern is usually represented graphically for the far-field conditions in either horizontal or vertical plane. It shows the variation of the field intensity of an antenna or antenna array as an angular function with respect to the axis.

The antenna array radiation pattern we took here is at one time, one polarization, and one plane cut. Since the patterns are usually presented in polar or rectilinear form with a dB strength scale, patterns are normalized to the maximum graph value, 0 dB, and the directivity is given for the specific antenna array. For simplicity, the radiation pattern we will show in this chapter is just a simple plot result from the data, because the pattern will automatically be scaled up or down by the tool, thus it shows the approximate shapes at different times. Noting that the radiation pattern can change dramatically depending upon frequency, time, and the wavelength to antenna characteristic length ratio, since almost all antennas are designed for a particular frequency range and usually the characteristic length is a multiple of $\lambda/2$ minus 2-15% depending on specific antenna characteristics. While for UWB antennas, the radiation pattern will change significantly depending on the time as well.

Antenna gain is assumed to mean directional gain of the antenna compared to an isotropic radiator transmitting to or receiving from all directions, and the -3dB beamwidth is a measure of the directivity of the antenna. Polarization, which is the direction of the electric (not magnetic) field of an antenna, is

another important antenna characteristic. Although we will not consider this factor too much in this work, this will still be a good consideration when optimizing reception or jamming.

Below is a typical radiation pattern in frequency domain for a horn from the literature:

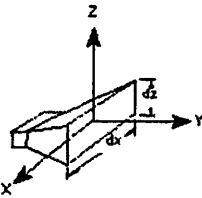
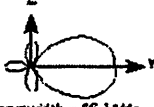
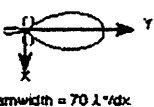
Antenna Type	Radiation Pattern	Characteristics
<p>HORN</p> 	<p>Elevation: </p> <p>3 dB beamwidth = $56 \lambda / dz$</p> <p>Azimuth: </p> <p>3 dB beamwidth = $70 \lambda / dx$</p>	<p>Polarization: Linear</p> <p>Typical Half-Power Beamwidth: 40 deg x 40 deg</p> <p>Typical Gain: 5 to 20 dB</p> <p>Bandwidth: If ridged: 120% or 4:1 If not ridged: 67% or 2:1</p> <p>Frequency Limit: Lower: 50 MHz Upper: 40 GHz</p>

Figure 3.4 Radiation pattern and characteristics for a Horn antenna

The radiation pattern is a very important antenna characteristic. Although it is usually most convenient and practical to measure the pattern in the receiving mode, it is identical, because of reciprocity, to that of the transmitting mode. Reciprocity for antenna patterns is generally provided by the materials used for the antennas and feeds, and when the media of wave propagation are linear. Nonlinear devices such as diodes, can make the antenna system nonreciprocal.

The antenna can be of any shape or size, and they do not have to be matched to their corresponding feed lines or loads, provided there is a distinct single propagating mode at each port. The only other restriction for reciprocity to hold is that the antennas in transmit and receive modes should be polarization matched, including the sense of rotation. This is necessary so that the antennas can transmit and receive the same field components, and thus total power. If the antenna that is used as a probe to measure the fields radiated by the antenna under test is not of the same polarization, then in some situations the transmit

and receive patterns can still be the same. For example, if the transmit antenna is circularly polarized and the probe antenna is linearly polarized, there is a way to represent the pattern of the circularly polarized antenna in either the transmit or receive modes. During this procedure, the power level and sensitivities must be held constant.

However, in our system, we should be very careful when using this theorem of reciprocity. We need to clarify the definition of our “relative angle” reciprocity first. Although the polarization of the transmitter and receiver do match with each other when TX is held directly towards the RX, as RX rotated we can still drive TX radiation pattern by looking at the RX side (given the assumption that theorem of reciprocity holds). There exists *significant* difference between the situations like Figure 3.5. We define “relative angle reciprocity holds” only when given the relative Azimuth/Elevation angle between TX and RX, e.g. 40°_{az} in Fig 3.5, the received radiation pattern will be the same, and thus, uniformly implies the TX radiation pattern as a function of *relative angle* between TX and RX in stead of a function of both TX angle and RX angle. For narrow band antennas, this will definitely be true, since the radiated waveform doesn’t depend on TX/RX angles, but same statement is no longer valid in our measurements, since the polarization of the transmitter and receiver doesn’t match any more and the waveforms will be a function of the angles. This gives a clue to the question raised at Chapter 2 for reciprocity, and implies we need to describe radiation patter as a function of both TX and RX angle.

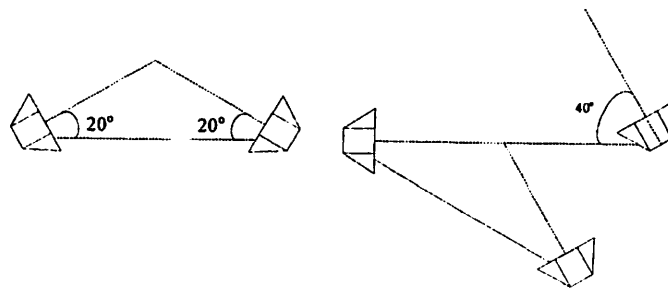


Figure 3.5 Antenna rotating and illustration of reciprocity

3.2.2 Single Antenna Response and Radiation Pattern

As we described when setting up the system, the ideal antenna we want to use is an omni-directional UWB antenna, therefore, we take a first set of measurements to have an idea of what our single antenna beamwidth is and how the radiation pattern looks like.

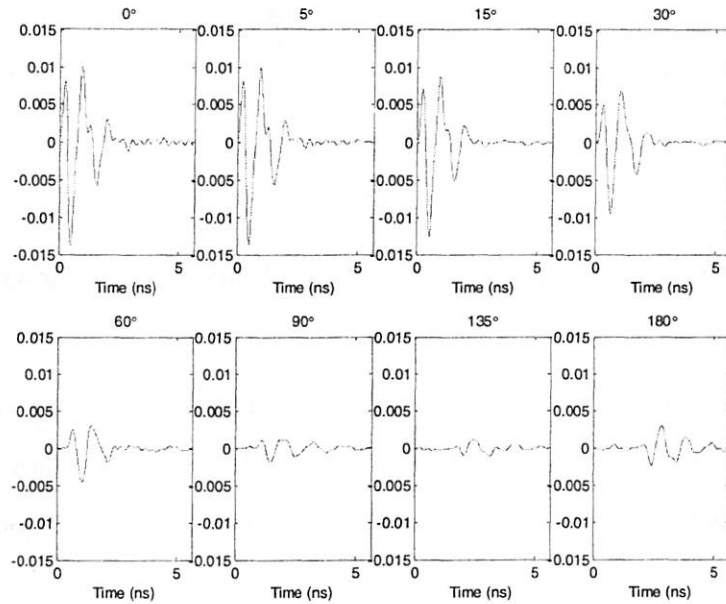


Figure 3.6 System response for some typical RX angles when TX angle=0

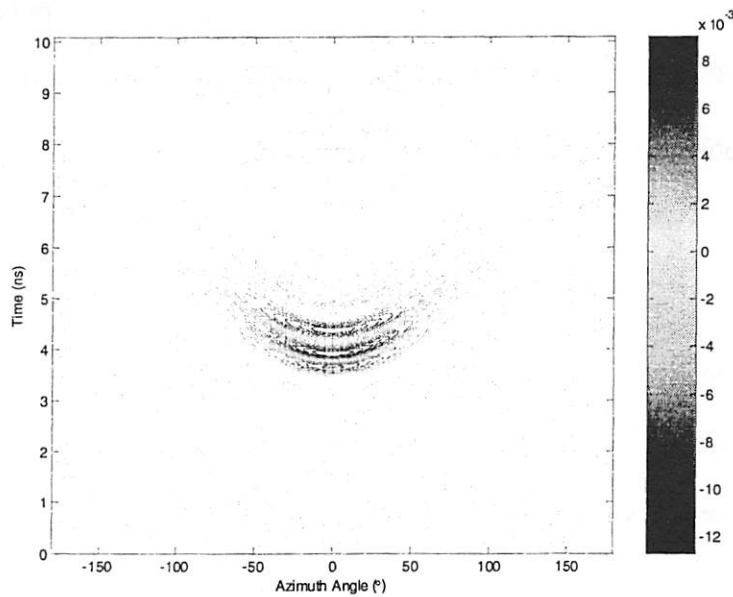


Figure 3.7 a) Measured Time vs Azimuth Angle plot for single antenna

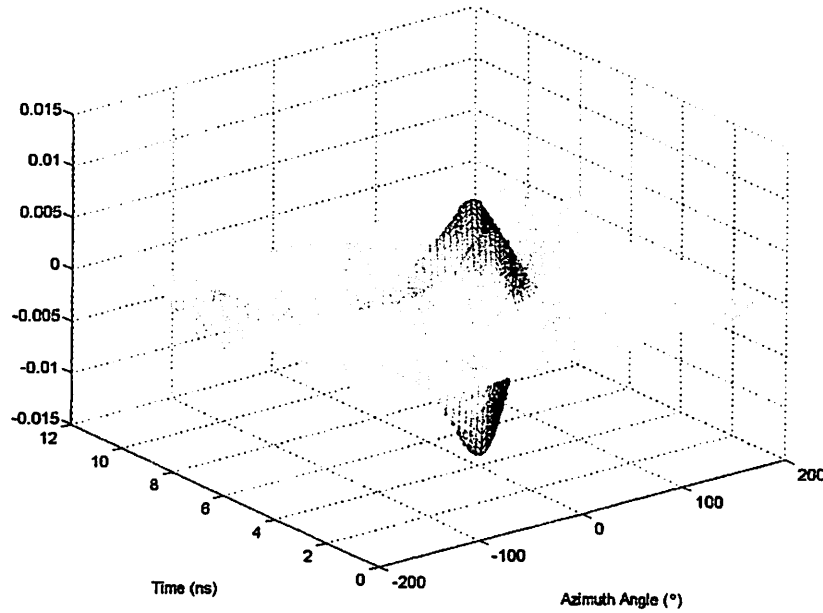


Figure 3.7 b) 3-D Plot of measured signal vs Azimuth Angle for single antenna

The above two plots show the time domain waveforms of the received signal as a function of Azimuth angle, when we fix our TX and let the horn face directly towards the RX (which is what is defined as zero degrees in the system).

To get the transmitter and receiver response, first notice that the received waveform is the system response composed of four parts: pulse generator $p_t(t)$, transmitter antenna $a_t(\phi, t)$, receive antenna $a_r(\phi, t)$, and oscilloscope $p_r(t)$. Suppose $h(\phi, \varphi, t, \tau)$ denote the channel response, then the received signal is

$$r(\phi, t) = \iiint h(\psi, \varphi, t, \tau) s(\phi - \psi, \varphi, \tau) d\tau d\varphi d\psi$$

where the system response $s(\phi, \varphi, t)$ is a composition of

$$s(\phi, \varphi, t) = p_r(t) * a_r(\phi, t) * a_t(\varphi, t) * p_t(t)$$

But the measured response is

$$m(\phi, t) = p_r(t) * a_r(\phi, t) * a_t(0, t) * p_t(t)$$

To find $a_r(\phi, t)$, we first plug the pulse generator directly into the oscilloscope and measure $p(t) = p_r(t) * p_t(t)$. Attach the horn antennas at both transmitter and receiver. In Chapter 2, we already found the close-in or near field distance, d_0 is smaller than 5m, so we need to make sure the transmitter receiver separation $r > d_0$. Then we can measure the response $m_0(\phi, t)$. This is not exactly the response $m(\phi, t)$, if the length of the antenna positioner arm l is negligible compared to the distance r .

Correction can be done $m(\phi, t) = m_0(\phi, t - d(\phi)/c)$, where c is the velocity of light. Denote $d(\cdot)$ is given by $d(\phi) = \sqrt{(l+r)^2 + l^2 - 2(l+r)l \cos \phi}$, if we can assume $a_r(\phi, t) = a_r^*(\phi, t)$, taking the Fourier transform on $m(\phi, t)$ with respect to t , yields

$$M(\phi, f) = P(f)A_r(\phi, f)A_r^*(0, f) = \frac{1}{j2\pi f} P(f)A_r(\phi, f)A_r(0, f)$$

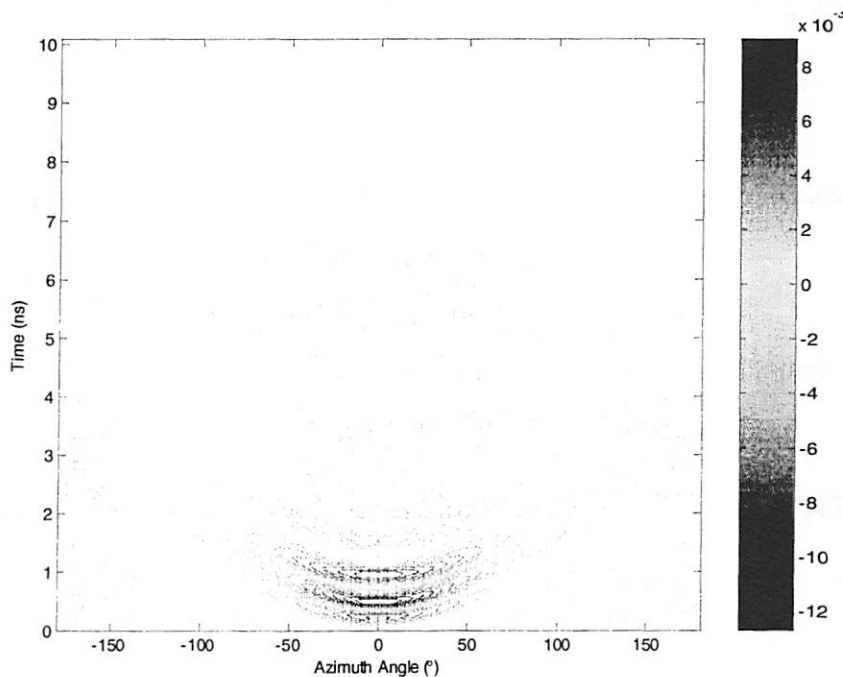


Figure 3.8 a) Corrected Time vs Azimuth Angle plot for single antenna

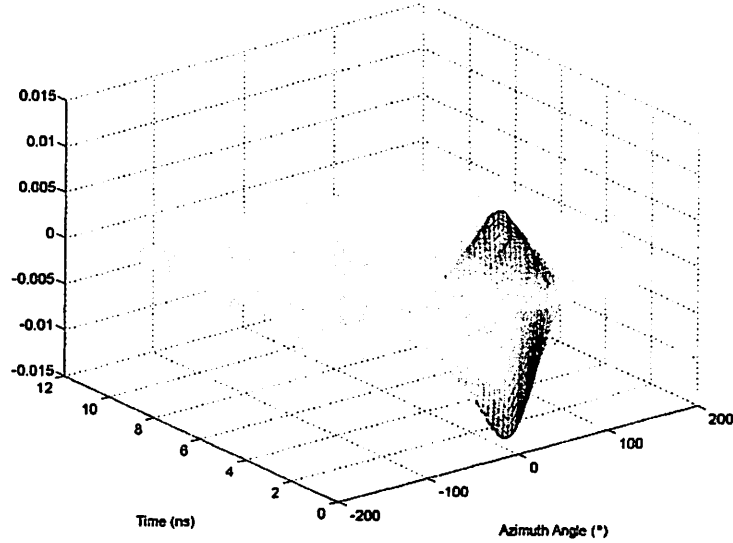


Figure 3.8 b) 3-D Plot of corrected signal vs Azimuth Angle for single antenna

At $\phi = 0$, we have $M(0, f) = \frac{1}{j2\pi f} P(f) |A_r(0, f)|^2$. This implies that

$$|A_r(0, f)| = \sqrt{\frac{j2\pi f M(0, f)}{P(f)}}$$

Substituting back into the $M(\phi, f)$, we obtain

$$a_i(\phi, t) = FT^{-1} \left\{ \frac{j2\pi f M(\phi, f)}{P(f) A_r(0, f)} \right\}$$

Then we can calculate the system response for that particular antenna

$$s(\phi, \varphi, t) = FT^{-1} \left\{ \frac{A_r(\varphi, f) M(\phi, f)}{A_r(0, f)} \right\}$$

and the channel response can be written as

$$h(\phi, \varphi, t) = \sum_i \alpha_i \delta(\phi - \psi_i) \delta(\varphi - \varphi_i) \delta(t - \tau_i)$$

We get TX antenna response and RX antenna response as Fig. 3.9 a) and b)

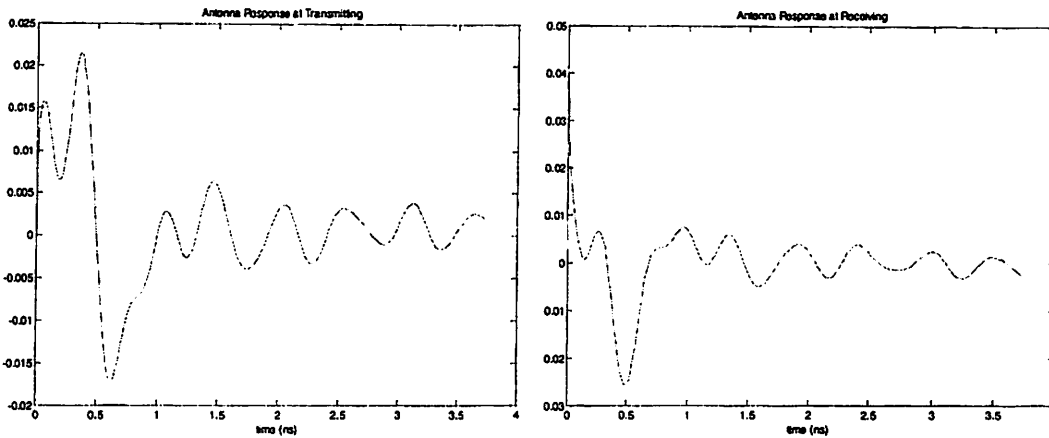


Figure 3.9 a) TX antenna response b) RX antenna response

And we get radiation patterns at different times looking like Fig. 3.10.

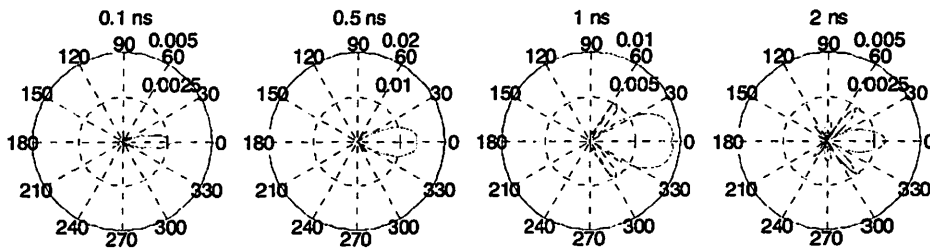


Figure 3.10 Radiation pattern at different times for single antenna

3.2.3 RX side and TX side radiation pattern

In this section, first we are going to investigate two categories of arrays on the receiver side, one of which is short/long array, whose design purpose is to investigate how the distance between two antennas affects the system response and the radiation pattern; the other set is linear/cross/rectangular array, each with four antennas, and we will see later how the beamwidth and time spreading is traded among these three types of arrays. Secondly we combine the information and get transmitter side radiation pattern.

3.2.3.1 RX side radiation pattern

- *Short/Long Array*

Manipulating the distance between two antennas and see how the system response changes. Measured system response and 3-D time vs Azimuth angle plot are shown.

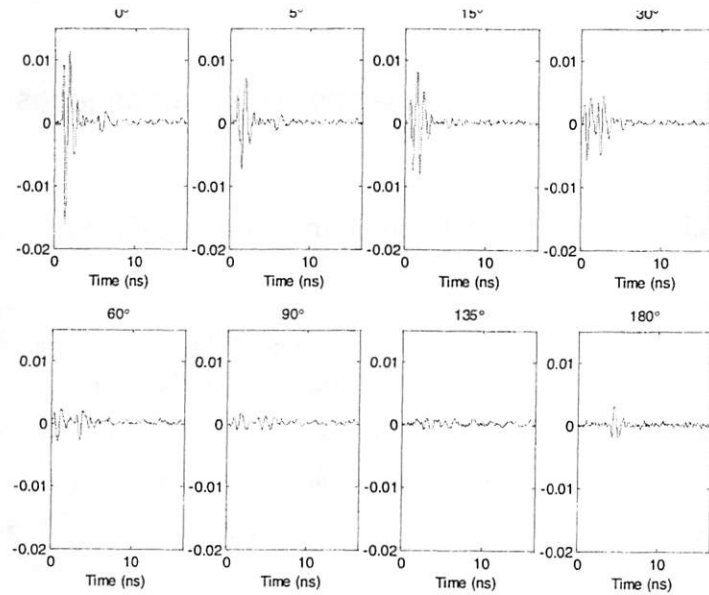


Figure 3.11 System response at different angles for short array

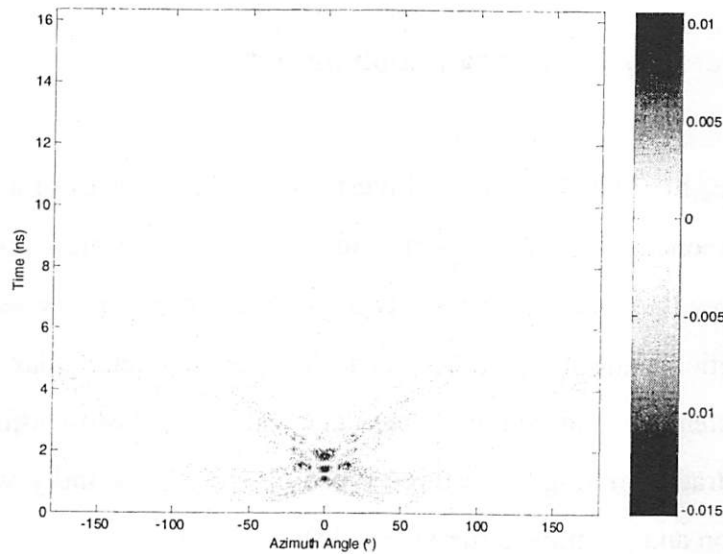


Figure 3.12 a) Measured Time vs Azimuth Angle plot for short array

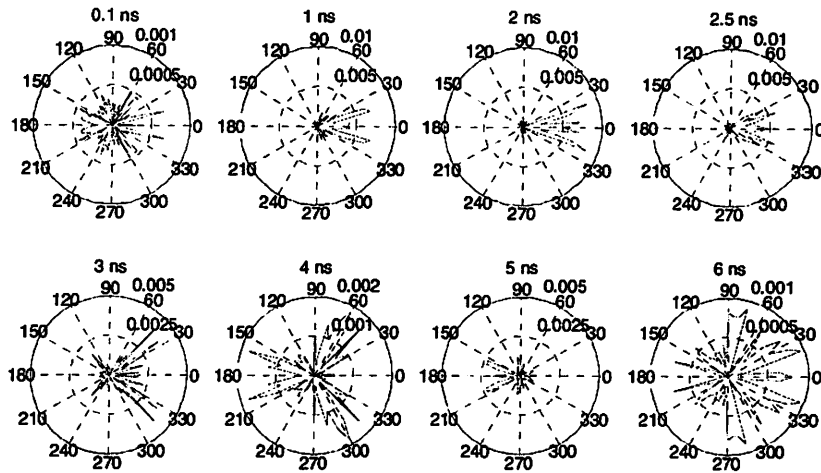


Figure 3.12 b) Radiation patterns at different times for short array

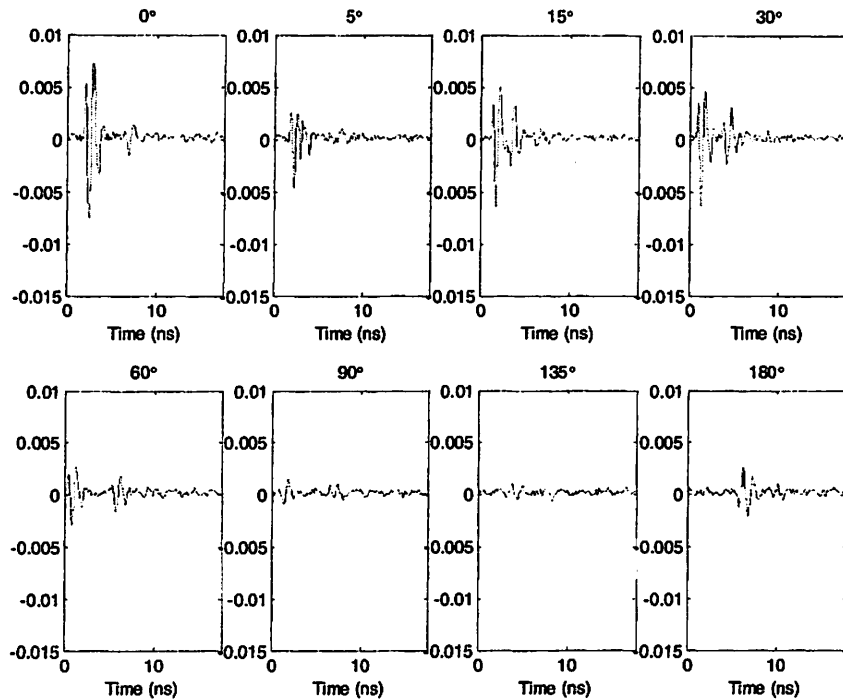


Figure 3.13 System response at different angles for long array

Horizontally separating two antennas further and keep polarization the same as short array, comparing the system response and radiation pattern in Figure 3.13 and 3.14, the trade off between beamwidth and time spreading can be summarized as follows: When the antennas are far more separated, the radiation pattern plot of the most top right shows that the beamwidth is around

10 degree, which is nearly one half of that of short array. This advantage brings us the benefit that the receiver is more oriented towards the TX, therefore, lowering the number of possible existing interferences from that certain direction. On the other side, the time spreading of the waveform is much more significant than the short array, and this long tail oscillation may mixing up with the multipath reflection in the real channel, thus diminish the positive energy enhancement brought in by the multipath effect.

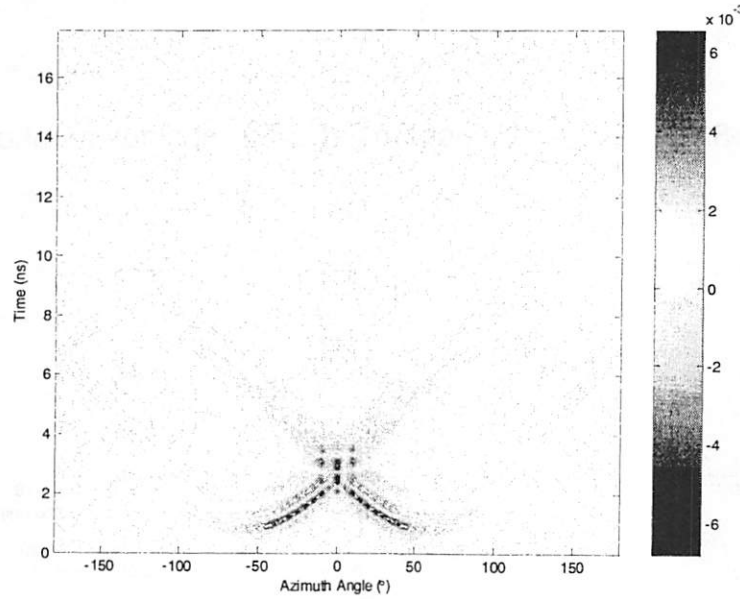


Figure 3.14a) Measured Time vs Azimuth Angle plot for long array

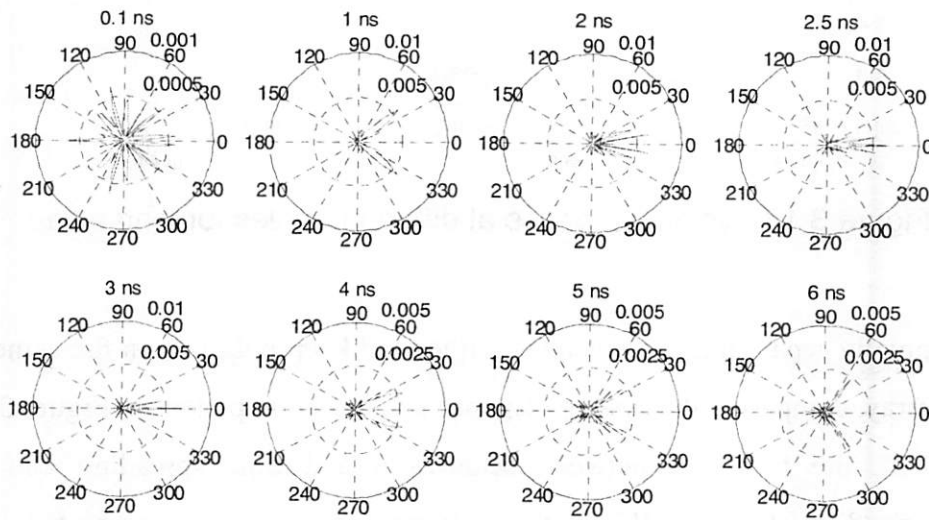


Figure 3.14 b) Radiation patterns at different times for long array

- *Linear/Cross/Rectangular Array*

While fixing the distance between each pair of antennas, arrange the antennas to form an array of either linear, or cross, or rectangular, we explore the after-correction system response and the radiation pattern as following.

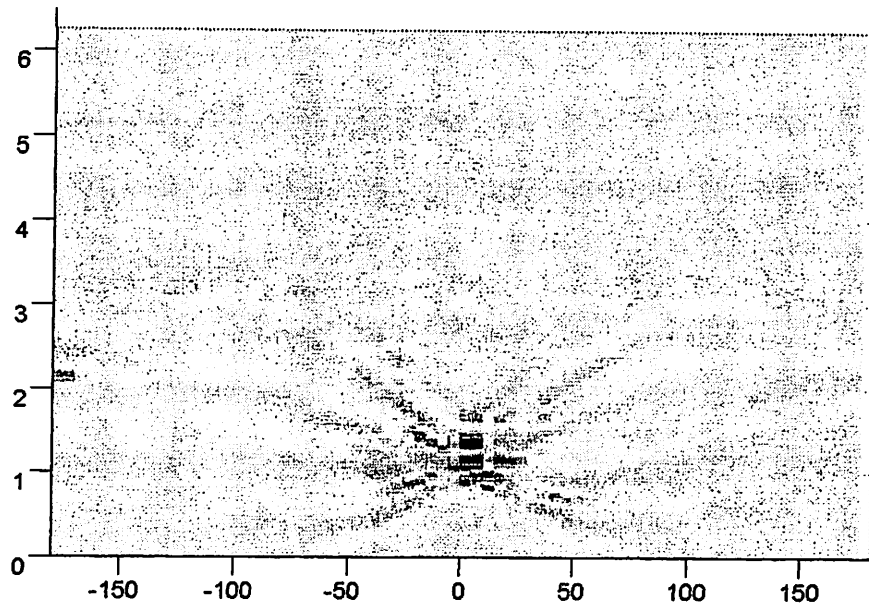


Figure 3.15 a) After-Correction Time vs Azimuth Angle plot for linear array

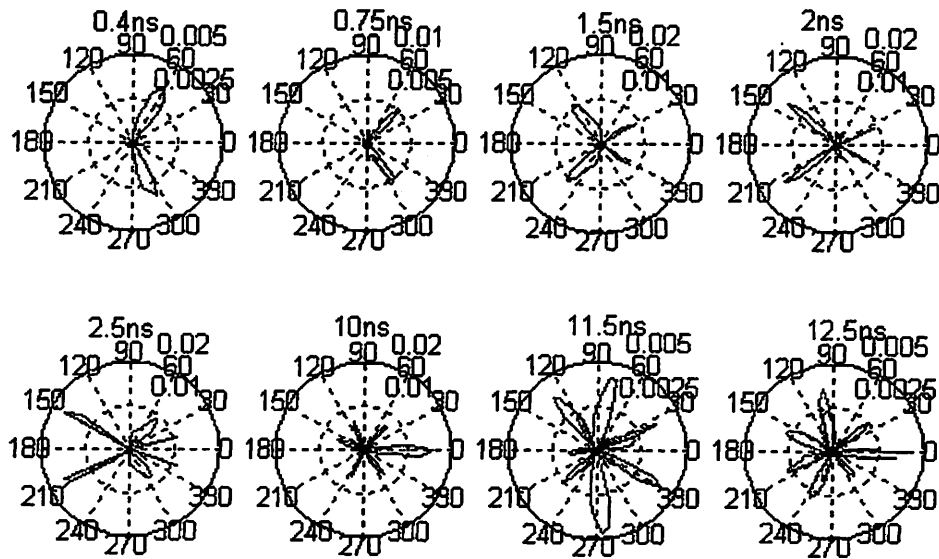


Figure 3.15 b) Radiation patterns at different times for linear array

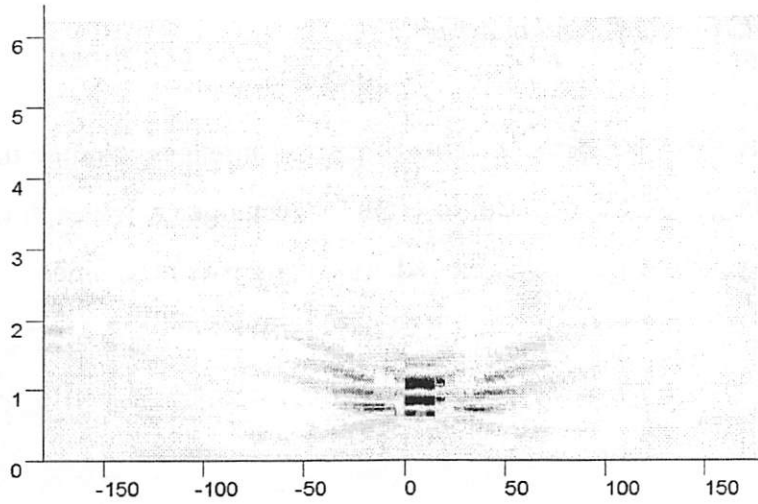


Figure 3.16 a) After-Correction Time vs Azimuth Angle plot for cross array

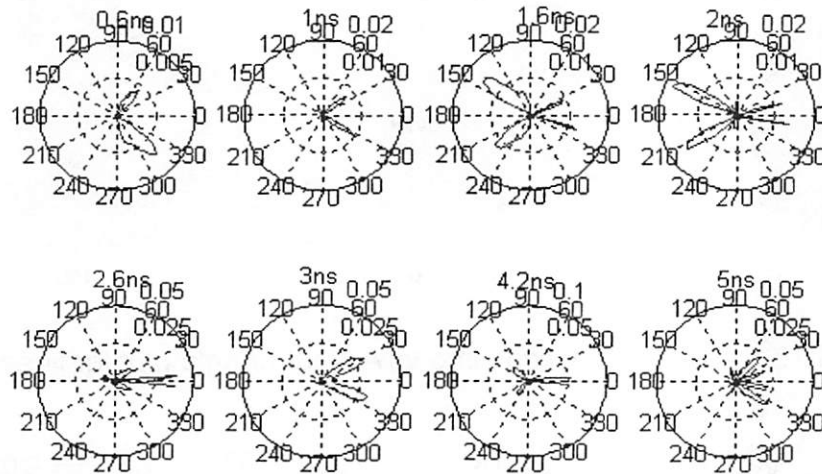


Figure 3.16 b) Radiation patterns at different times for cross array

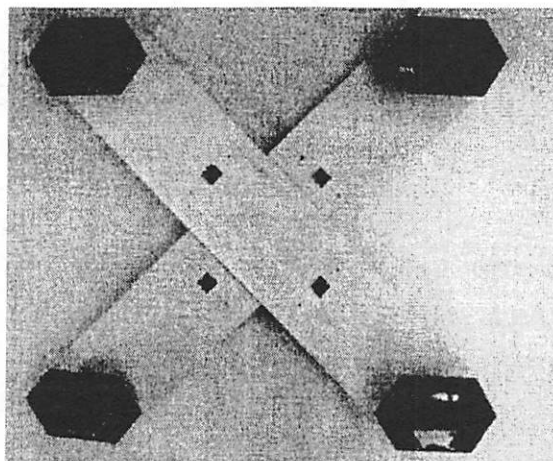


Figure 3.17 Rectangular array with 4 antennas and tongue at bottom

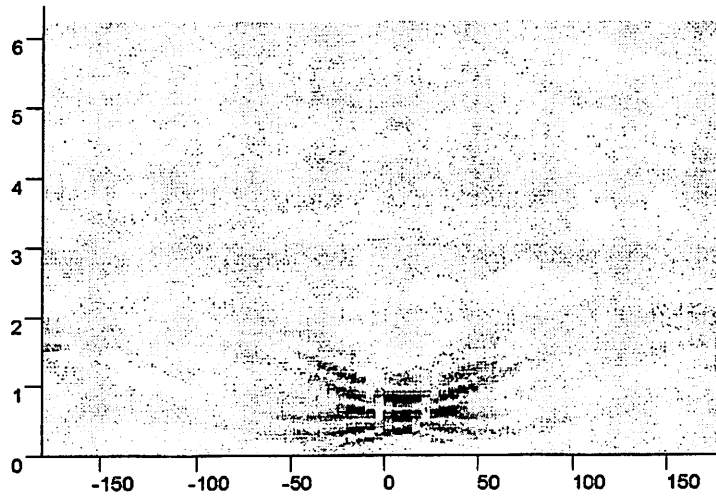


Figure 3.18 a) After-correction time vs Azimuth angle plot for rectangular array

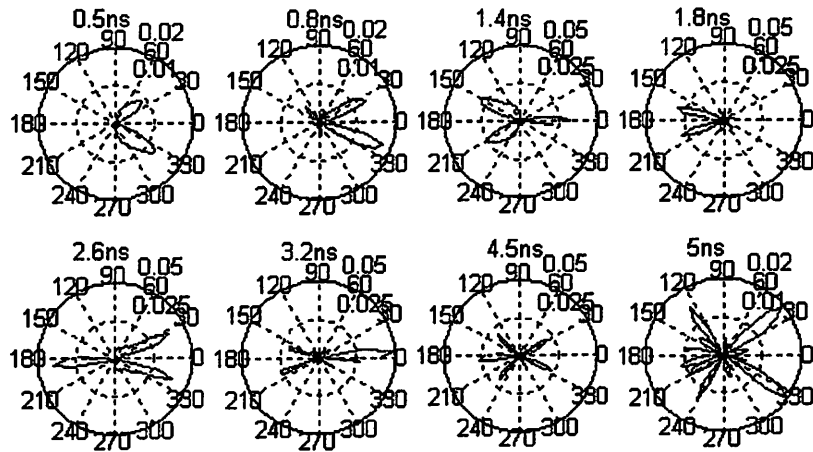


Figure 3.18 b) Radiation patterns at different times for rectangular array

Focusing on part a) of each Figure of 3.15, 3.16, 3.18, time spreading is narrower in the sequence of linear, cross and rectangular. Now let's compare the peak point radiation pattern in Figure 3.19

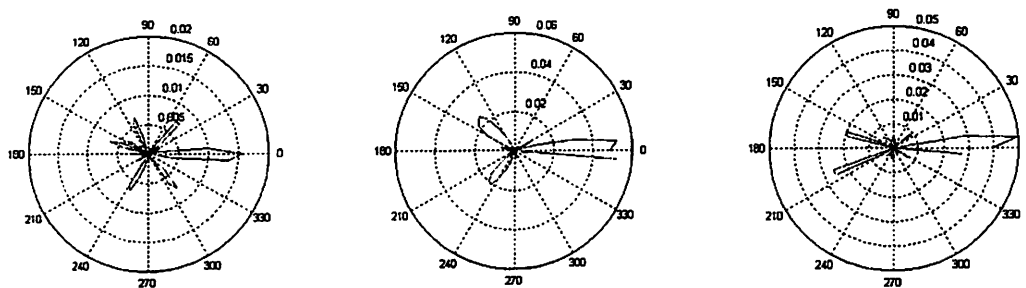


Figure 3.19 a) Linear array b) Cross array c) rectangular array

In spite that there existing some none zero offset for the measurement of the last two sets of arrays, we can still clearly find out that the beamwidth is sharper in the reverse sequence of linear, cross, and rectangular. Thus we further proof that there is a significant trade off between time spreading and beamwidth, which limits our accuracy if we design our system to satisfy some directionality and vice versa.

3.2.3.2 TX side radiation pattern

Recap that our single antenna has beamwidth of $\pm 30^\circ$, to achieve the goal of making our transmit antenna omni directional by rotating it, we test two sets of radiation patterns, based on rotating the TX 30-degree each step (Figure 3.20) and 20-degree each step (Figure 3.21). When transmitter is facing 0 degree and receiver is rotating 30 degree each step, by adding up 12 signals aligned at same time spot with synchronization and correction of the extra time caused by the arm of array, the time domain waveform looks like Figure 3.22 and time versus azimuth angle 3-D plot like Figure 3.23.

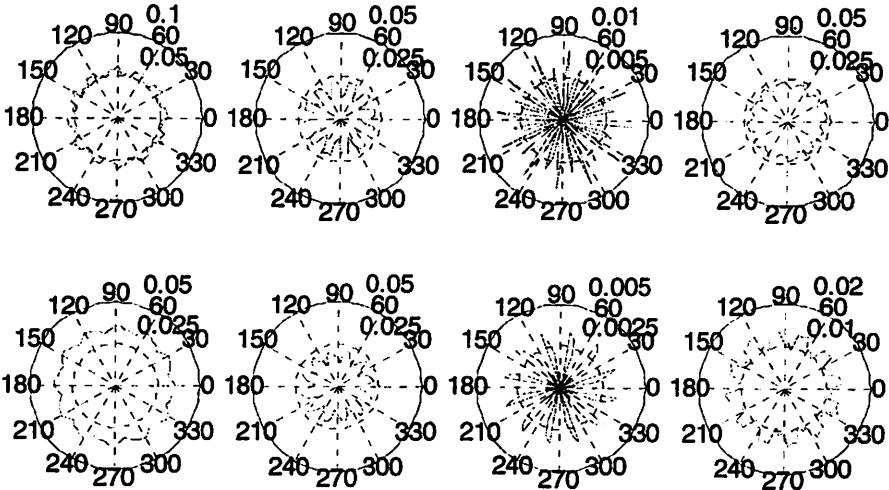


Figure 3.20 Transmitter radiation pattern for rotating 30 degree each step

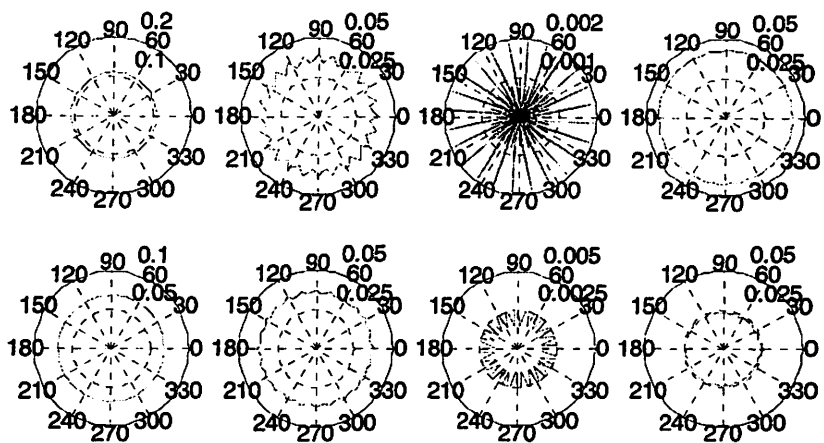


Figure 3.21 Transmitter radiation pattern for rotating 20 degree each step

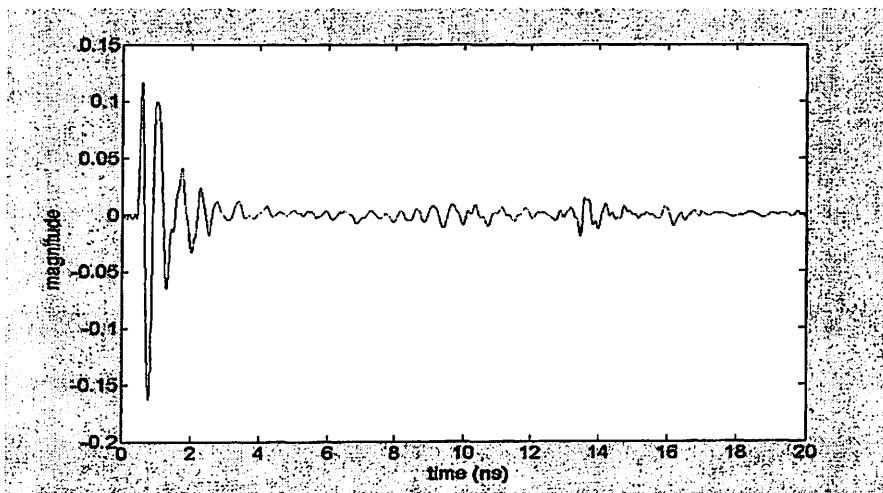


Figure 3.22 Added waveform when TX facing 0°, RX rotating 30° each step

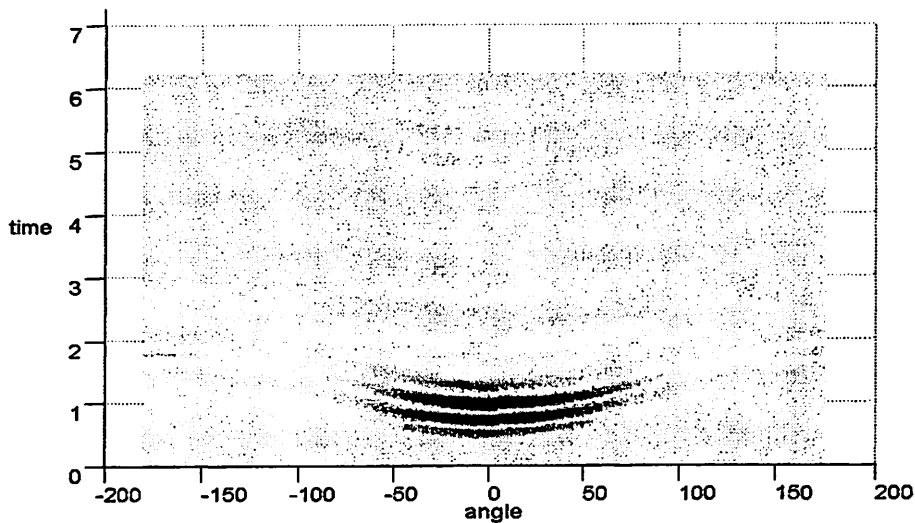


Figure 3.23 Time vs Azimuth angle plot for TX

Clearly, 20-degree step size is more omni-directional compared to 30-degree with round circle and very little peaking phenomena at times, so ideally during the channel measurement, we will make the TX rotating 20 degree each step.

Quick note here for estimation of the tolerable measurement time when making TX resolution 20 degree, RX resolution 5 degree, pulse generator frequency 100KHz. t_1, t_2, t_3, t_4, t_5 will be defined in the attachment.

Average No.	256	1024	2048	4096
t1 (s)	1161	1219	1291	1440
t2 (s)	120	120	120	120
t3 (s)	10	10	10	10
t4 (s)	10	10	10	10
t5 (s)	120	120	120	120
Total	4.30 h	4.50h	4.74 h	5.23 h

Table 3-1 Tolerable measurement time

3.2.4 Receiver Sensitivity

All receivers are designed for a certain sensitivity level based on requirements. Sensitivity in a receiver is normally taken as the minimum input signal S_{\min} required to produce a specified output signal having a specified signal-to-noise ratio (SNR) and is defined as the minimum signal-to-noise ratio times the mean noise power,

$$S_{\min} = (S/N)_{\min} KT_0 B(NF)$$

In which, K is Boltzmann's Constant, 1.38×10^{-23} Joule/°K, T_0 is Absolute temperature of the receiver input (°Kelvin), B is receiver bandwidth and NF is noise figure.

For a signal impinging on the antenna, the system level sensitivity is known as minimum operational sensitivity (MOS),

$$S_{\min} = (S/N)_{\min} KT_0 B(NF) / G$$

Where G is the system or antenna gain. Since MOS includes antenna gain, it may be expressed in dBLi (dB referenced to a linear isotropic antenna). When specifying the sensitivity of receivers intended to intercept and process pulse signals, the minimum pulse width at which the specified sensitivity applies must also be stated. Thus, we have a lower MOS if temperature, bandwidth, NF, or S/N_{\min} decreases, or if antenna gain increases. For communications and commercial broadcasting receivers, sensitivity is usually stated in micro-volts or dBv.

One would not design a receiver with more sensitivity than required because it limits the receiver bandwidth and will require the receiver to process signals it is not interested in. In general, while processing signals, the higher the power level at which the sensitivity is set, the fewer the number of false alarms which will be processed. Simultaneously, the probability of detection of a "good" low-noise signal will be decreased. It is defined as the ratio of the response, which will be a negative number with more negative being better sensitivity.

The SNR in a receiver is the signal power in the receiver divided by the mean noise power of the receiver. All receivers require the signal to exceed the noise by some amount. Usually if the signal power is less than or just equals to the noise power it is not detectable. For a signal to be detected, the signal energy plus the noise energy must exceed some threshold value. Therefore, just because N is in the denominator doesn't mean it can be increased to lower the MOS. SNR is a required minimum ratio, if N is increased, then S must also be increased to maintain that threshold. The threshold value is chosen high enough above the mean noise level so that the probability of random noise peaks

exceeding the threshold, and causing false alarms, is acceptably low. Fig. 3.24 shows the relationship between SNR and probability of detection and false alarm rate from the literature.

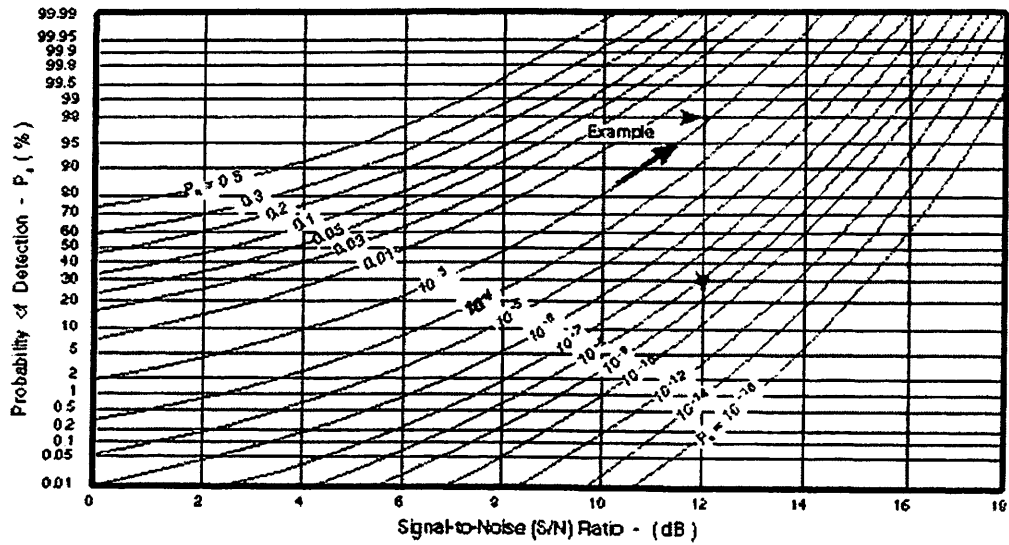


Figure 3.24 Nomograph of SNR ratio as a function of probability of Detection P_d and probability of false Alarm rate P_n .

Chapter4 Interference Effect and Cancellation

4.1 Introduction

In communication systems, the timing and phase of the transmitted signal must be recovered at the receiver. It is also necessary for optimum reception to measure the channel. For analog data measurements, the receiver must estimate the transmitted waveform at finely spaced sample times, however, the existence of interference complicates the recovery problem a lot.

In this chapter, we are going to look at first several time and frequency domain signal processing techniques which are candidates for our analysis tool-set of the received signals, and then, based on the method we chose, signals are processed along with interference at different positions and from different antennas.

Spectrum occupancy, related to cognitive radio applications will be given from the measurements made in the Berkeley downtown area, from which we can see the spectrum is not crowded in reality. A method of canceling/mitigating the interference is proposed at the end, which may give us hints to some future cognitive radio interference cancellation research work.

4.1.1 Power Spectrum Density Estimation

Given an arbitrary real zero mean wide sense stationary process $\{Y(t); t \in \mathfrak{R}\}$

and suppose we want to estimate its power, $K_Y(0) = E[Y^2(t)]$, from a sample function of duration T. If we know nothing about the process other than the fact that it is zero mean and wide sense stationary (WSS), then a reasonable estimation is given by

$$P_T = \frac{1}{T} \int_0^T Y^2(t) dt$$

We can see this estimation is **unbiased** for any choice of $T > 0$ by taking the expectation and interchanging it with integration,

$$E[P_T] = E[Y^2(t)]$$

Next we want to show that the estimation is **consistent** by showing that the variance of the error goes to zero as T approaches infinity. To show this, at least for the case of a WSS zero mean Gaussian process, consider $E[P_T^2]$, we have

$$E[P_T^2] = \frac{1}{T^2} \int_0^T \int_0^T E[Y^2(t)Y^2(\tau)] dt d\tau$$

Noting that for Gaussian random variables, it is true that

$$E[Y^2(t)Y^2(\tau)] = [E[Y^2(t)]]^2 + 2[E[Y(t)Y(\tau)]]^2 = K_Y^2(0) + 2K_Y^2(t - \tau)$$

thus we have

$$E[P_T^2] = [E[P_T]]^2 + \frac{1}{T^2} \int_0^T \int_0^T 2K_Y^2(t - \tau) dt d\tau$$

$$VAR[P_T] = \frac{1}{T^2} \int_0^T \int_0^T 2K_Y^2(t - \tau) dt d\tau$$

since the quantity inside the integral is non-negative for all t, we can upper bound it by integrating over all t, so

$$VAR[P_T] \leq \frac{1}{T^2} \int_{\tau=0}^T \int_{t=-\infty}^{\infty} 2K_Y^2(t) dt d\tau = \frac{2}{T} \int_{t=-\infty}^{\infty} K_Y^2(t) dt$$

which means if $\int_{t=-\infty}^{\infty} K_Y^2(t) < \infty$, then the variance in the estimate P_T goes to 0 as

$1/T$ [Gallager Notes], the proof can be obtained even for non-Gaussian processes.

The reason why we want to frame our problem with the wide sense stationary assumption is that for the power estimation problem, in the real world, most of the processes can be regard as wide-sense stationary. Our channel characterization measurement process is another WSS example. A “good” estimation, which is inherently unbiased and consistent, can provide a view of how to get a reasonable estimation of power spectrum density. So in the following, we are going to walk step by step to find the closest approximation of the actual power spectrum density given the observation of the signal.

The first option is to use a narrow band filter with input X , we see easily that $S_x(\omega_0) \approx \frac{\pi}{\epsilon} E[P_T]$, given

$$S_y(\omega) = S_x(\omega); \quad ||\omega - \omega_0| \leq \epsilon/2$$

$$= 0; \quad \textit{elsewhere}$$

and a good estimate of $S_x(\omega_0)$ is $\hat{S}_x(\omega_0) \approx \frac{\pi}{\epsilon} P_T$, the variance of the estimation error is small when T is large. However, given the ultra wide band signal sending, this will be not really implementable, since the higher the accuracy of the estimation, the longer the data sample will be required, whereas for short windows we will lose some information of the power spectral density.

The second option comes from looking at the autocorrelation function. The estimation of an actual autocorrelation is

$$\hat{K}_x(\nu, T) = \frac{1}{T-\nu} \int_0^{T-\nu} X(t)X(t+\nu)dt$$

and if we define $A(\omega, T) = \int_0^T X(t)e^{-j\omega t} dt$, then

$$\hat{S}_x(\omega, T) = \frac{1}{T} A(\omega, T)A^*(\omega, T) = \frac{1}{T} \int_0^T \int_0^T X(t)X(\tau)e^{-j\omega(t-\tau)} dt d\tau$$

$$= \frac{2}{T} \int_{\nu=0}^T \int_{\tau=0}^{T-\nu} X(\tau)X(\tau+\nu)e^{-j\omega\nu} d\tau d\nu$$

put in the estimation of autocorrelation, we get

$$\hat{S}_x(\omega, T) = 2 \int_0^T \left(1 - \frac{\nu}{T}\right) \hat{K}_x(\nu, T) e^{-j\omega\nu} d\nu$$

This will be a poor estimate of PDF of X even for large T. The problem is that the error variance in $\hat{K}_x(\nu)$ is proportional to $1/T$, but these errors are being integrated in $\hat{S}_x(\omega, T)$ over an interval of size T. Thus it is not surprising that the error in $\hat{S}_x(\omega, T)$ is not decreasing with T.

Now, if one still wants to get a reasonable estimate of PDF from a finite segment sample function of X, the modified Fourier transform approach is

$$\hat{S}_x(\omega, T) = 2 \int_0^T \alpha(\nu) \hat{K}_x(\nu) e^{-j\omega\nu} d\nu$$

where $\alpha(\nu)$ is a function of ν , but not T, that has the value one until ν gets large and then goes to zero fast enough so that $\int_0^{\infty} \alpha(\nu) d\nu < \infty$, this function can get rid of the noise in the $\hat{K}_x(\nu)$ for large ν , and serves to smooth $\hat{S}_x(\omega, T)$ over a small range of ω in roughly the same way that the frequency analyzer of the first approach smoothes over frequency. In the following session, we will compare different PSD estimation problems, built and evolved from this model.

There are mainly three categories of estimation available for power spectral density analysis in the MatLab tool: non-parametric, parametric and subspace method. The first, non-parametric method is going to be focused with some brief introduction to the latter two, since we are going to work on the PSD from the received signal itself directly.

The simplest way is to use “periodogram”, which returns the power per unit frequency, and is good for high SNR and long data sequence, although it is not a consistent estimator. We has the option to choose what kind of window to be

for the estimator, e.g. Hamming, Kaiser, Chebwin. The FFT size used is the maximum of 256 or the one we defined, e.g. 100K. Fig 4.1 shows the periodogram plot for the same signal, by using three different windows. We see by changing window type, the spectrum leakage problem is greatly solved, however, obviously, the tradeoff between resolution and variation can be evaluated to make the decision on which window to use.

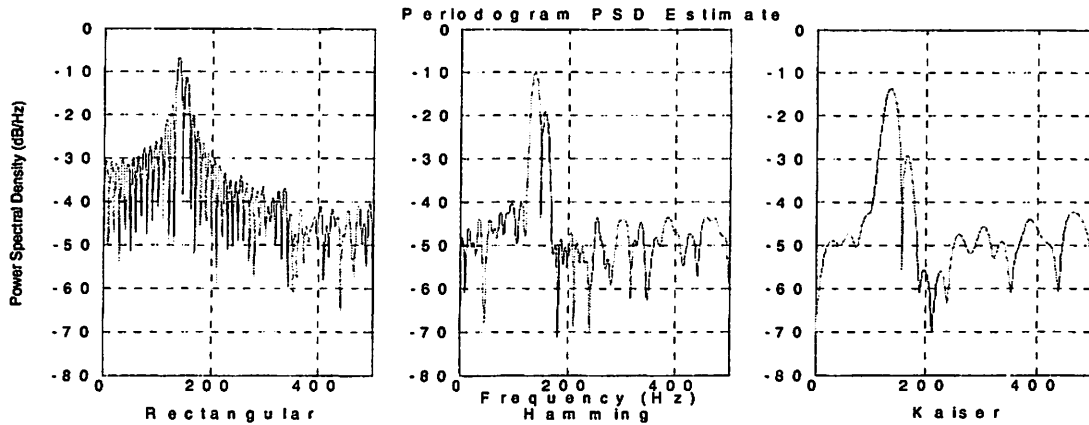


Figure 4.1 Periodogram plot for Rectangular, Hamming and Kaiser Window

Improved way for estimation is to use “Welch”, which divides the data into some overlapping segments, computing a modified periodogram of each segment, then average through all the overlapping sets. The window type is still flexible and the percentage of the overlap can be defined by the user.

It improves the performance of the estimation of low SNR signal quite a bit when a good balancing point is given from the tradeoff between the resolution and variance, noticing the variance is inversely proportional to the number of segments. Fig.4.2 compares the “Welch” and “Periodogram” when using the same window and the PSD of “Welch” itself with different window types.

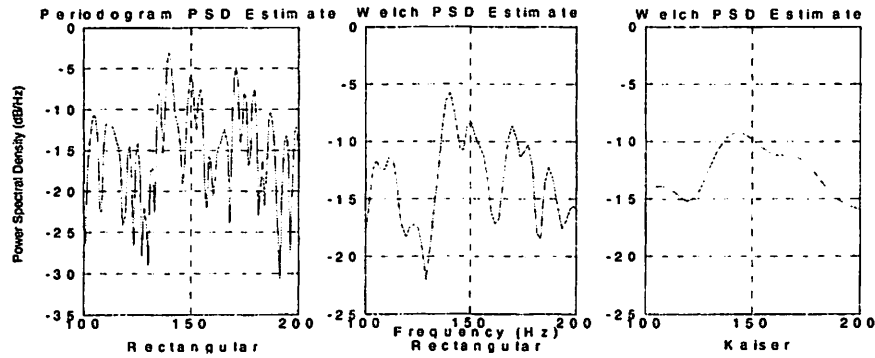
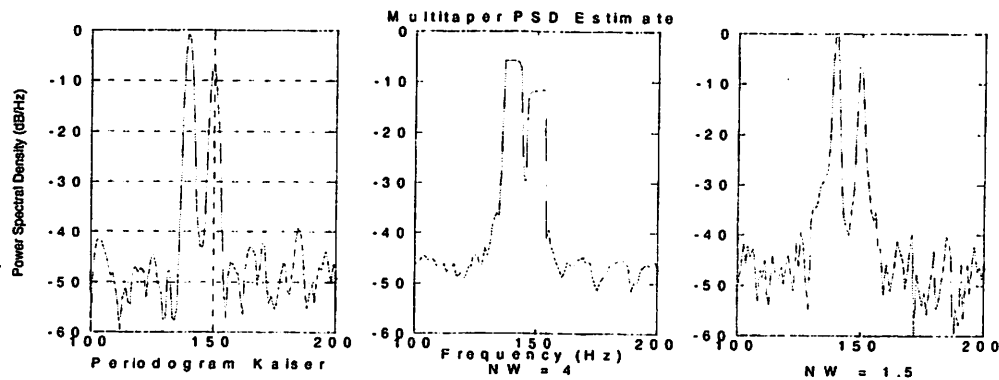
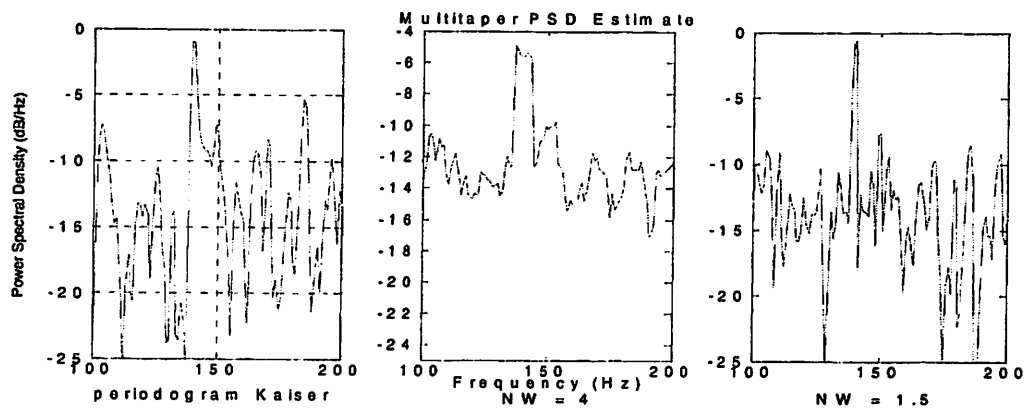


Figure 4.2 Periodogram and Welch plot for Rectangular window and Welch plot for Kaiser Window

A modern way to solve this estimation is the so called “Multitaper” method, which filters the signal through a filter bank of L optimal FIR filters derived from discrete prolate spheroidal sequences (DPSS). The parameter of time-bandwidth product “ NW ” controls the resolution and variation tradeoff: as NW increases, the variance decreases and BandWidth for each taper increase. Although having nice performance and being able to estimate both high and low SNR cases, it turns out to be a pretty time consuming technique, especially for the UWB and the long duration signal like ours. One solution is to save all the generated DPSS in a MAT-file for the future use, so it simplifies to a one-time-consuming problem. Fig 4.3 shows under the high SNR/low SNR conditions, the comparison between the “Multitaper” and the “periodogram” using Kaiser window, and presents the effect of the “time-bandwidth product” on the PSD.



a) High SNR



b) Low SNR

Figure 4.3 High/Low SNR: Periodogram plot for Kaiser window and Multitaper plot for Time-BandWidth Product = 4 and 1.5

Now we introduce the parametric methods and subspace methods. When the data is relatively short duration and the PSD is assumed to be the output of a linear system driven by white noise, and we are supposed to estimate the parameters/coefficients of the linear system which hypothetically "generates" the signal, parametric methods could give better performance than the non-parametric methods. For example, the Yule-Walker autoregressive (AR) method which is an autoregressive (AR) spectral estimate of a time-series from its estimated autocorrelation function and the Burg method and Covariance method, which minimize the linear prediction errors and the forward prediction errors.

The subspace method is called, a high resolution and super resolution method, which has self control signal number and threshold. It is an effective detection method for low SNR and can detect the spectra of sinusoidal signals buried in noise. Two typical examples are the multiple signal classification (MUSIC) method and Eigenvector (EV) method, which are pseudo spectrum estimation based on eigen analysis of the autocorrelation matrix.

Finally, in our problem of estimation to the interference, we are going to use more the windowing “Welch” method, since it matches the accuracy of our system without bringing too much complicated math work, and the result is pretty accurate.

4.1.2 Deconvolution Method

Channels can be characterized by their transfer function in the frequency domain or by their impulse response in the time domain. Here we are going to show the existing deconvolution methods for both frequency and time domain. Each of them has its advantage in getting a clean output signal. However, all the measurements under investigation were done in the time domain, so the time domain methods are more preferable. Deconvolution of the time-domain waveforms can be used to determine the impulse response of a linear time-invariant system. And the indoor channel is assumed to be time-invariant is the transmitter and the receiver are static and no motions take place in the channel and more theoretically, the signal should be at least wide sense stationary during all the time of measuring.

The essential reason for deconvolution comes from the limited bandwidth of available test signals as compared to the bandwidths of the channels themselves. After deconvolution, channel will be modeled by resolving the impulse response components into bins smaller than the duration of the sounding pulse and the reciprocal of the channel bandwidth.

If $h(t)$ is the impulse response of such a channel whose input is $X(t)$, then the output $y(t)$ is given by the convolution integral,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t - \tau) d\tau$$

In the frequency domain, the convolution transforms into multiplication, and ideally, deconvolution can be performed in the frequency domain using the Fourier transformation, thus do

$$H(j\omega) = Y(j\omega) / X(j\omega)$$

except at the tagged frequencies, at which $H(\omega)$ is set to zero. To be more explicit, we see that due to measurement and signal processing limitations, simple division around the zeros of $X(j\omega)$ will result in noise-like error. Filtering should be used to improve the estimation of the impulse response. A filter that demonstrated quality performance is given by the form

$$H(j\omega) = \frac{1}{1 + \lambda / |X(j\omega)|^2} \cdot \frac{Y(j\omega)}{X(j\omega)}$$

and we can see from some previous measurements that $\lambda = 50$ will give a nice result for the corresponding impulse response.

Another available way is to use the FIR estimator deconvolution method. By this way, we match the channel frequency response with a MP FIR filter by zero replacement. The resulting filter is phase corrected and channel equalization is performed

$$H_F(j\omega) = e^{-j\omega\tau_b} \sum_{k=0}^{M-1} b_k e^{-j\omega_k}$$

And the above filter transfer function has the form of a FIR filter whose tap weighs are the filter-impulse response as given by

$$h_F(n) = \begin{cases} b_n, & \text{when } 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases}$$

compared with $H_F(j\omega) = \sum_{k=0}^{N-1} \alpha_k e^{-j\omega_k}$

we just make $\tau_b = 0, M = N, b_n = a_n$ and arbitrary τ_k is replaced by a

sampled value k , the channel can be then represented by a FIR filter, $H_F(j\omega) \equiv H(j\omega)$, thus $h_F(t) \equiv h(t)$.

The impulse response is not really a summation of delayed delta functions in some sense, since in practice multipath components have different waveforms compared with the reference pulse. Though, the transfer function and the impulse response give full channel description, only few parameters can be used by the receiver for the channel estimation.

Advantage for frequency domain deconvolution is that the inverse filter is susceptible to high noise levels and its accuracy, which refers to the reconstructed channel response $r'(t)$ correlated with originally received signal $r(t)$ is high and gives a better initial reconstruction with correlation coefficient very close to 1.

Now we move to time domain deconvolution method. Mainly one, the subtractive deconvolution method, is going to be introduced here. And we take the approach to expand the algorithm, and call it the Multi-Template subtractive deconvolution method.

First of all, we recapture the model of the impulse response of the channel as the summation of effective scatters $h(t) = \sum_i^I a_i \delta(t - \tau_i)$, where a_i are the magnitudes of I scatters. The model is widely used and can adequately represent the channel for many narrowband communications purposes.

Subtractive deconvolution says, suppose the clean pulse $\bar{p}(t)$, which is the band-limited and distorted (by the receiver channel) form of $p(t)$, is the same

for all the responses for given antennas and front-ends. In the time domain we initialize the dirty map with $d(t) = r(t)$ and clean map with $c(t) = 0$, then use the clean pulse to find the peaks Γ_i in the dirty map and add them to the clean map $c(t) = c(t) + \Gamma_i \delta(t - \tau_i)$, then we minus it from the dirty map $d(t) = d(t) - \Gamma_i \bar{p}(t - \tau_i)$. Keep doing this until reaches the threshold, which is given by mixing the consideration of the environment and noise effect. Or, instead of subtracing it, we can also substitute a zero at the detected components if we assume the multipath components don't overlap. The flow chart is shown in Fig 4.4.

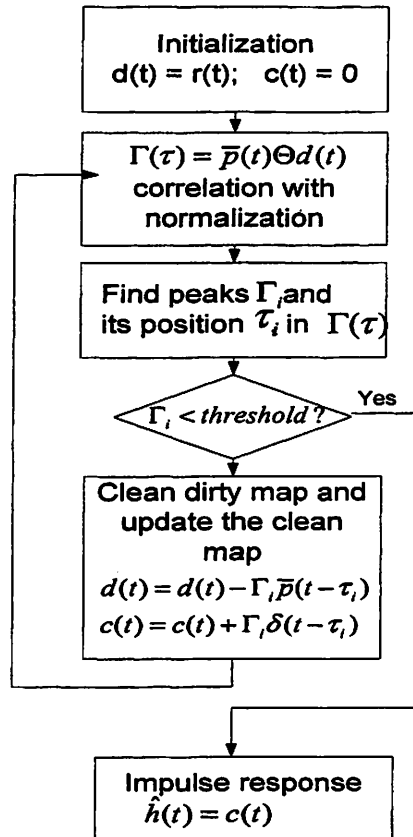
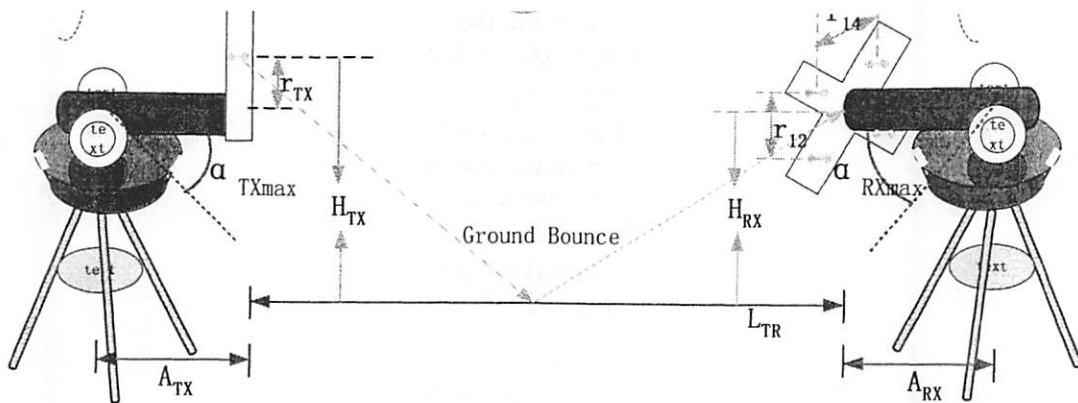


Figure 4.4 Flowchart for subtractive deconvolution method

Unfortunately, this model does not perfectly fit the UWB channel because the delta function at the receiver implies an infinite channel bandwidth, which is

not possible or realizable. To make the model more accurate, the reference pulse used is the convolution of the sounding pulse, transmitter and receiver antennas, and the impulse response of the detector (sampling oscilloscope). This reference pulse is measured in a well behaved channel at the Barn in Berkeley Downtown where the multipath reflections can be gated out. Setup of the system is that, as we mentioned at Chapter two, the transmitter and receiver antennas are facing each other with a distance of 6.24m to guarantee far field reception for the antenna under use. All the setup parameters and an example of the received pulses for ceiling bounce, ground bounce and LOS are shown in Fig 4.4 and Fig 4.5.



- Parameters

$$H_{TX} = 1.25m$$

$$H_{RX} = 1.25m$$

$$L_{TR} = 6.24m$$

$$A_{TX} = 0.20m$$

$$A_{RX} = 0.27m$$

$$r_{14} = 0.32m$$

$$r_{12} = 0.32m$$

$$r_{TX} = 0.10m$$

$$\alpha_{TXmax} = -35^\circ$$

$$\alpha_{RXmax} = -30^\circ$$

TX pulse

Ceiling bounce 35° — 35°

0-0 received data -- LOS

Ground bounce -20° — -20°

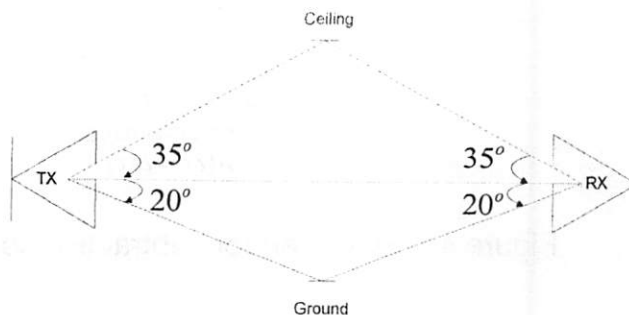


Figure 4.5 Measurement parameters for LOS, ceiling and ground bounce

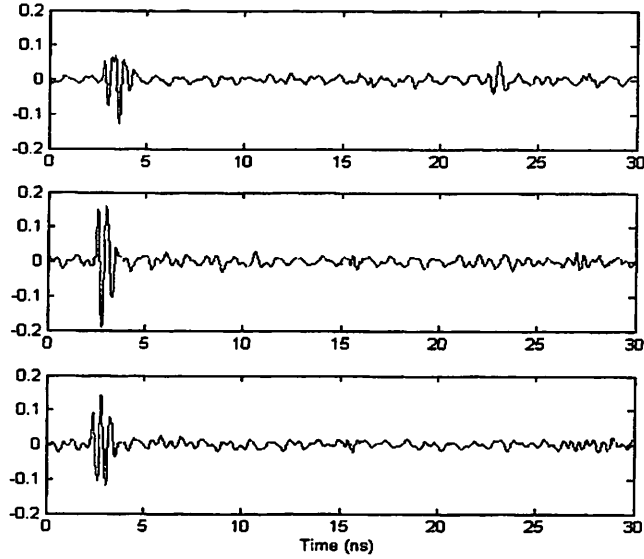


Figure 4.6 Ground bounce (top), LOS (middle), Ceiling bounce (bottom)

And we can see that the waveform associated with the LOS is considerably different from the reflected waveforms. On the other hand, for different angles the waveform is also different as we have shown in the previous chapter. So the improved subtractive deconvolution method, so called, Multi-template subtractive deconvolution algorithm is described below:

Given one TX angle θ_t , model the impulse response of the channel with a certain receiver angle θ_r as the summation of effective scatters

$$h(t, \theta_t, \theta_r) = \sum_i^I a_i \hat{h}^j(t - \tau_i)$$

where \hat{h}^j is the impulse response of the system, when excited by LOS pulse \bar{p}^1 , the output is the jth template \bar{p}^j , when the received signal is LOS, then it corresponds to $\hat{h}^1 = \delta(t)$. Still use the clean pulse $\bar{p}(t)$ to remove the distortion introduced by the receiver and the accuracy will be controlled by the threshold, which can be translated into SNR, e.g. choosing the threshold to be 0.05 of the maximum peak, means SNR is 26dB.

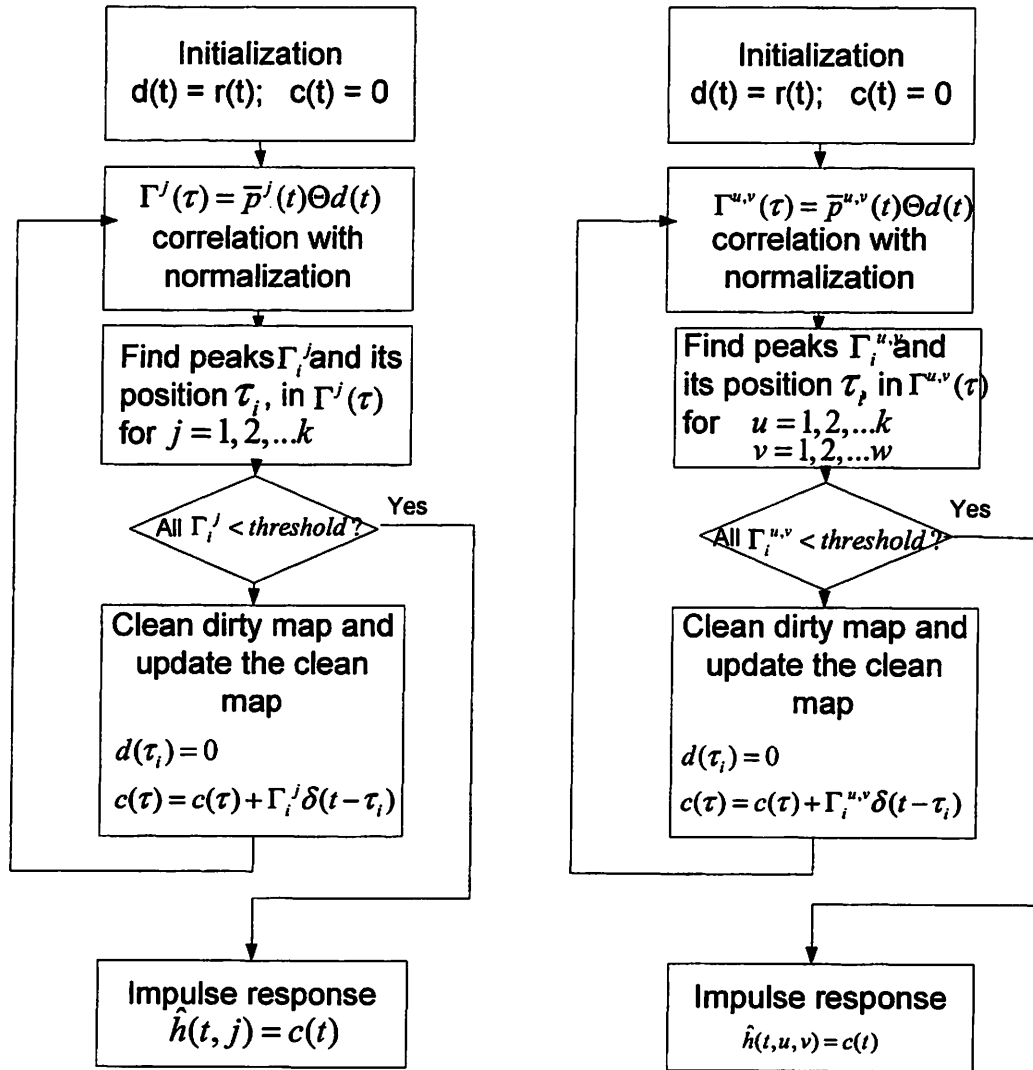


Figure 4.7 Flowchart for Multi-template subtractive deconvolution method

The time domain flowchart for multi-template subtractive deconvolution method is shown in Fig 4.7, opened the option for cleaning the dirty map with either minus or substituting zero.

Finally, when realizing multi-template deconvolution algorithm, following points are being considered.

- *Dynamic threshold is given after sensing the environment.*

- *Substituting zero, assuming non-overlapping multipath components, is more preferable than the “storage-needed minus components” way with notification of capturing most of the energy.*
- *Dimensions of the template goes up significantly when we need to do both azimuth and elevation angle of the TX and RX with four antennas. For this reason, the time domain deconvolution methods mentioned are only doable when we consider azimuth angle for the transceiver and a more sophisticated and efficient algorithm, such as the clean algorithm, may be required when dealing with a large dimensional signal.*

4.2 Interference at Different Positions

Measurements were taken at three different place, indoors, at the Berkeley Wireless research Center. These three positions are very typical, as one is in meeting room, whose interferences come with almost equal probability from inside and outside of the office; the second one is close to the wireless LAN transmitter, and third one is inside the lab, which may contain lots of signals coming from different equipment.

Three sets of measurements with time interval of one minute are taken in meeting room and two sets are taken at the other two positions, and the reason for more sets in meeting room is because we want to derive the clean map from the meeting room interference later. Dirty maps of interference for each position are plotted below in Fig. 4.8 to 4.10.

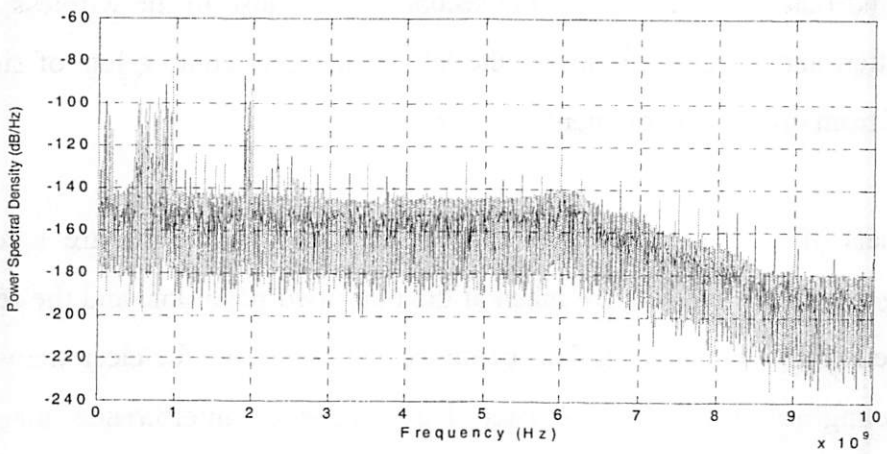
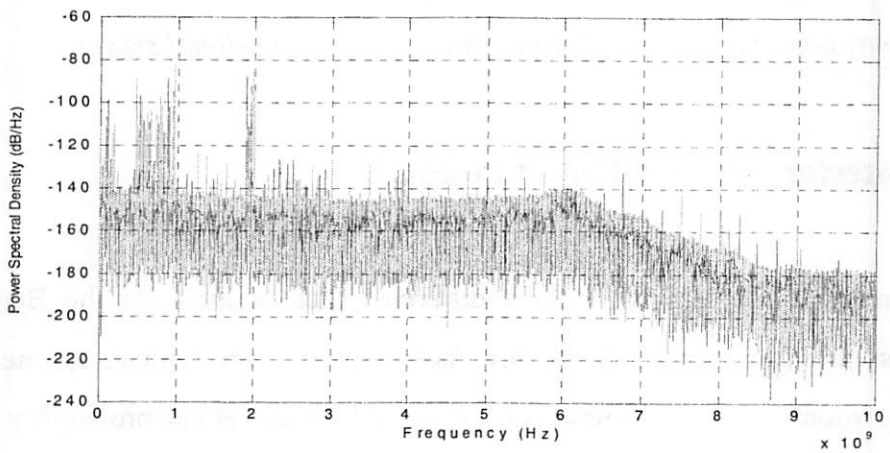
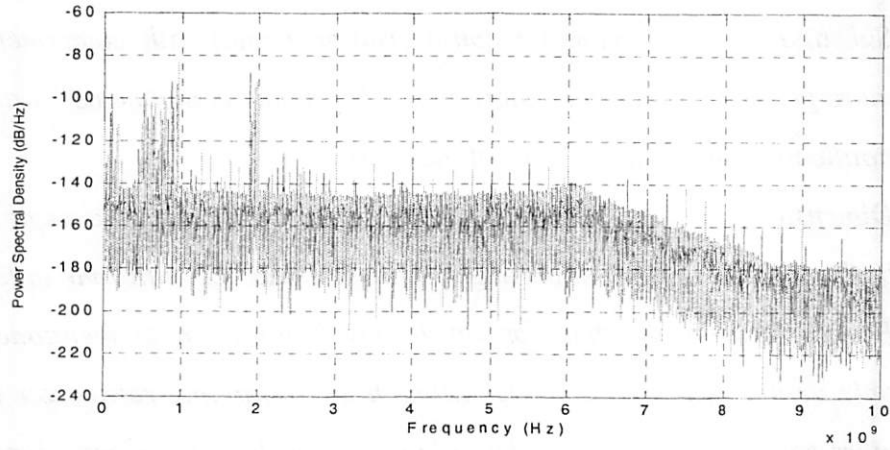


Figure 4.8 Dirty Map of interference at position in meeting room

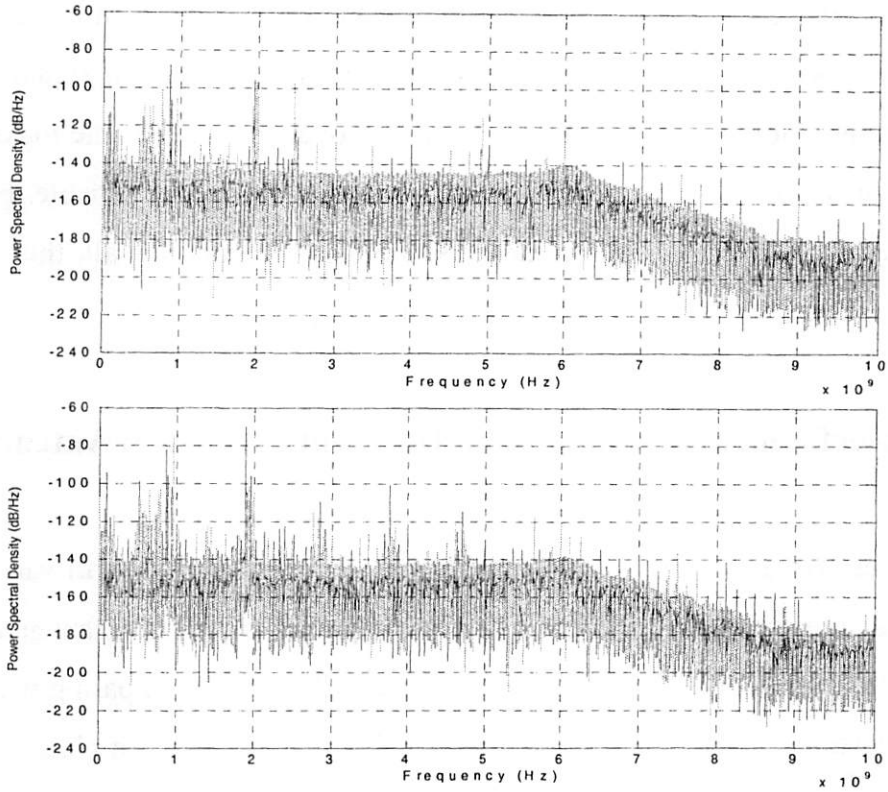


Figure 4.9 Dirty Map of interference at the wireless LAN transmitter position

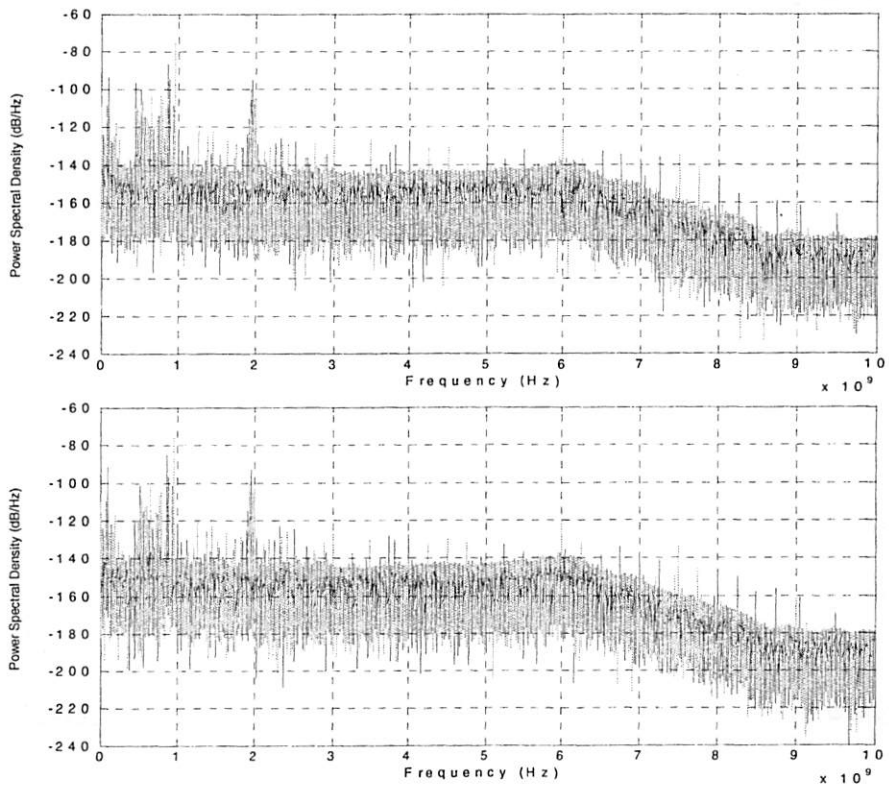


Figure 4.10 Dirty Map of interference at the Lab next to BEE position

However, if we take a closer look at the second dirty map taken close to the Wireless LAN, it is clear that there exists a big distortion when signal going through the 25dB amplifiers, and it is already over the limit of the maximum input that the oscilloscope can afford, so this PSD is no longer reliable. Except this, comparing the dirty maps, the noise floor is the same for all, thus proof that the interference is highly time and position dependent.

4.3 Interference Measured from Different Wideband Antennas

As we discussed when choosing the antennas, the directionality will vary with different antennas and a good example for comparing is the LLNL antennas we are using in most of the measurement and the TEM Horn. Comparing with the LLNL antenna, TEM horn is more directional, which means even placed at the same position, the received signal can be quite different as that of LLNL and the dirty frequency map of Fig 4.11 shows the difference.

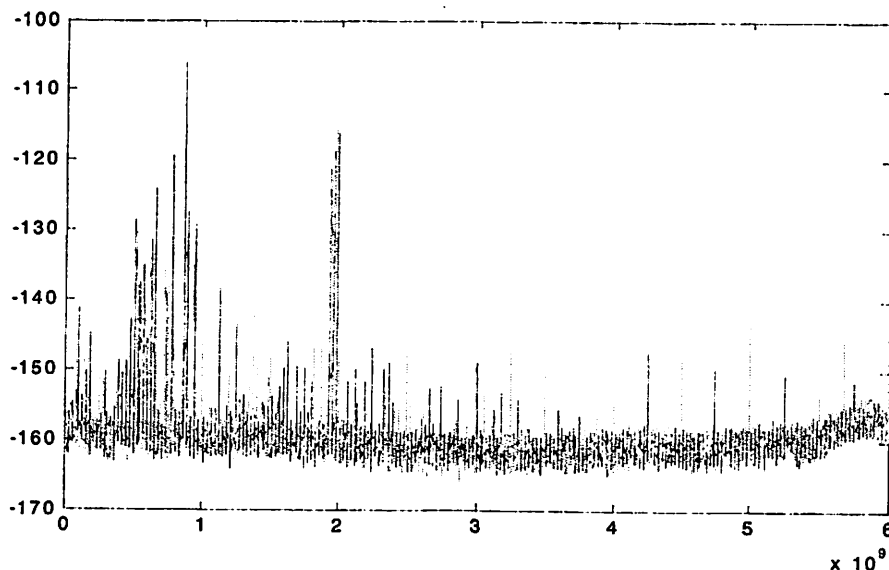


Figure 4.11 a) Dirty map ranged from 0 to 6GHz taken with LLNL

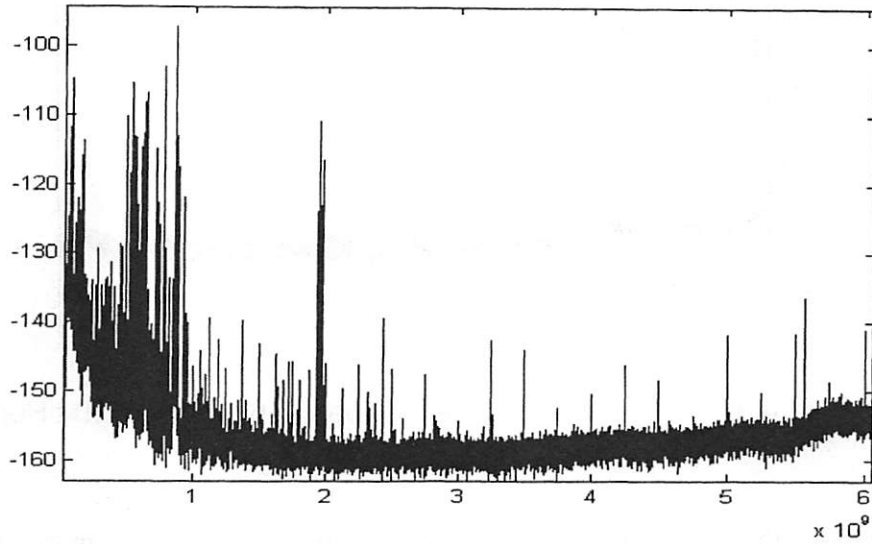


Figure 4.12 b) Dirty map ranged from 0 to 6GHz taken with TEM Horn

The noise floor for two antennas don't match perfectly, but even in this dirty map the frequency responses for the received signal have significantly dissimilarity and we are going to clean this map further and figure out the dissimilarities.

Before using the subtractive deconvolution method, noticing that we need to take in the fact that there exists harmonics and the equipment noise, the hardware noise for both cases as shown in Fig 4.12.

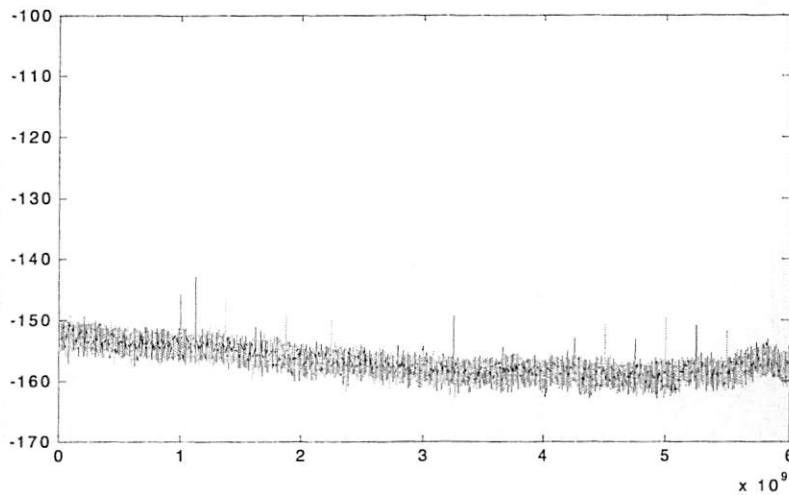


Figure 4.13 a) Equipment Noise between 0~6GHz with LLNL

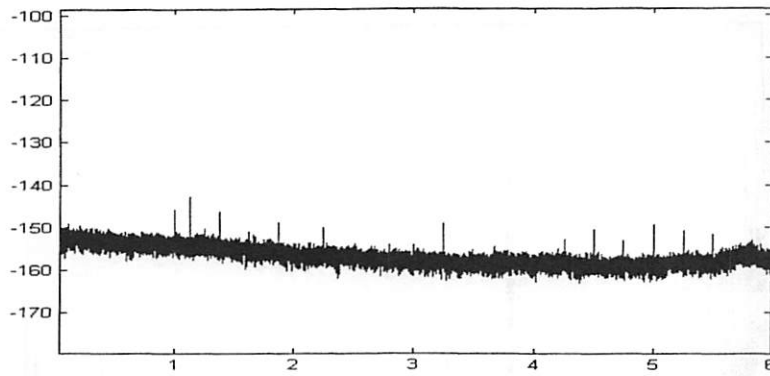


Figure 4.14 b) Equipment Noise between 0~6GHz with TEM Horn

Now using the powerful deconvolution method, the clean maps for the interference captured with LLNL antenna and the TEM horn are as follows, shown in Fig 4.13. The threshold refers to these clean maps is the one such that 95% of the total energy above the noise floor can be captured.

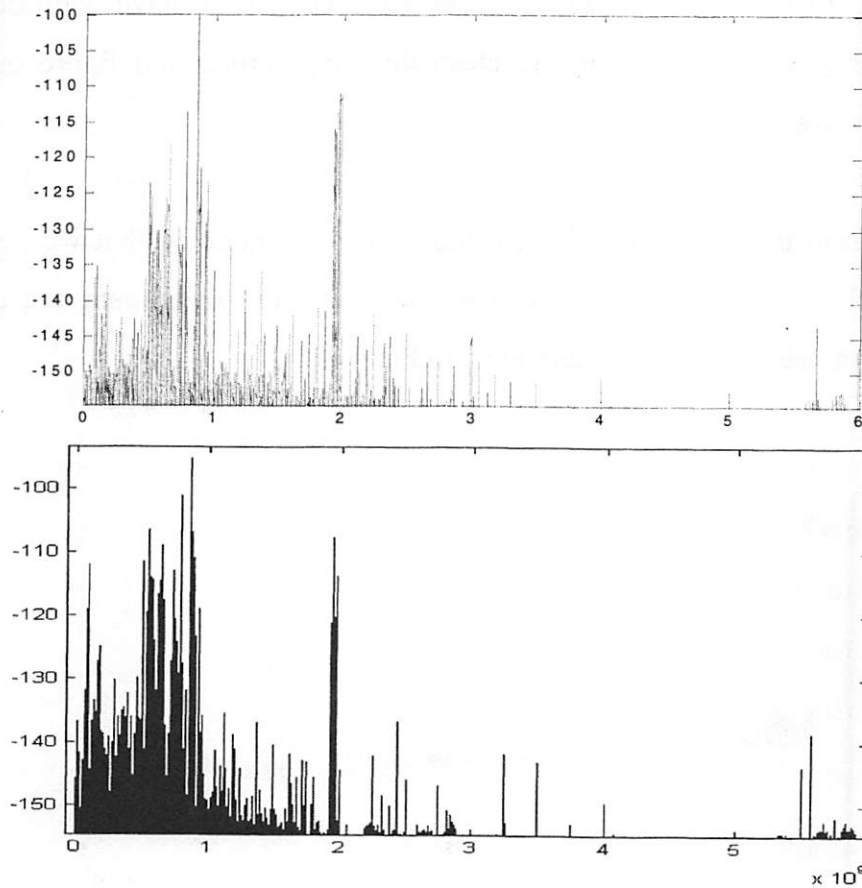
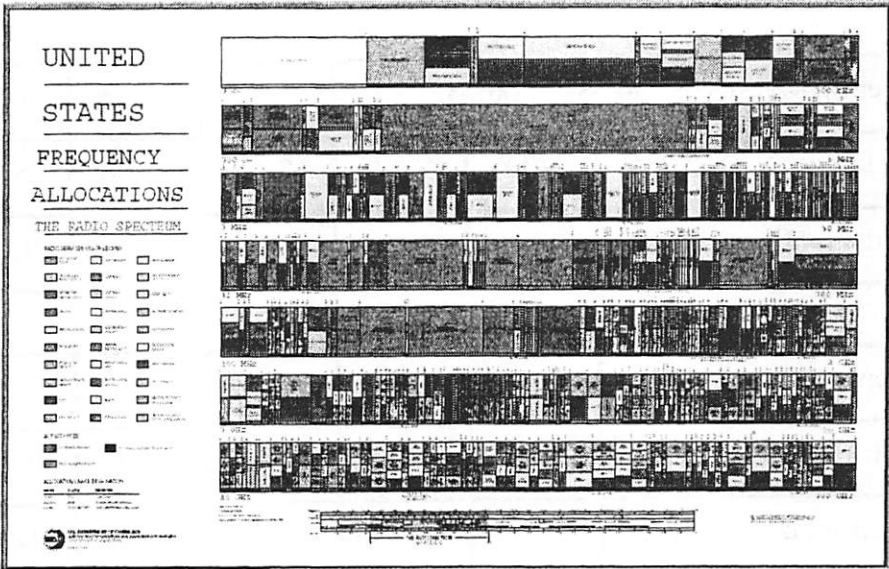


Figure 4.15 Interference map between 0~6GHz taken with a) LLNL b) TEM Horn

Comparing the two clean maps, both the magnitude and the frequency of interference change. If we can assume the position of two measurements is at one point (not exactly, but is approximately, if we regard the interference sources are far away), then these clean maps imply the directionality and the gain for two types of antennas. We may need to emphasis here that Figure 4.13 is the plot we are going to look at from time and time in the following sections and all analysis and proposals will be given under the realization of the clean interference map we got here.

4.4 Spectrum Occupancy and Current Spectrum Overview

From the existing spectrum policy of FCC shown on Fig 4.14, the spectrum behaves like a fragmented disk, most of which are already occupied. Good frequencies have been taken and we can imagine that the bandwidth is going to be more and more expensive if it follows this trend. However, there are two points in terms of improving the spectrum efficiency. First, more exploring on unlicensed bands will become one of the biggest innovations, and second will be dynamically sensing and making use of all the “empty” frequencies.



But how empty is the current spectrum? Recent measurement done by FCC shows that 70% of the allocated spectrum is not utilized and we are going to show from our measurements, at a typical position, how the current spectrum looks like and how it leads to the statement that the spectrum crisis exists more on paper than in reality with the idea of implementing “cognitive radio.”

Let’s review the interference map we got in Fig 4.13. Table 4.1 shows the usage percentage by looking at the map taken with the TEM Horn.

Frequency (Hz)	0~1G	1~2G	2~3G	3~4G	4~5G	5~6G
percentage (%)	55	35	10	0.438	0.212	6

Table 4-1 Usage percentage of spectrum taken in Downtown Berkeley

It is amazing that besides the 0 to 1GHz range, which is heavily used with occupancy percentage exceeds 50 percent, and the 1 to 2GHz range, which is slight heavy with PCS cell phones and some moderate amplitude signals, all other frequency ranges are very lightly used, especially for 3 to 5GHz range, with the occupancy percentage of 0.438%!!! Table 4.2 shows the position, bandwidth and magnitude of the existing signals from 3 to 5GHz.

Frequency(MHz)	3187.50	3250.00	3259.53	3500.00	3749.69	3773.80
BW (Hz)	280 K	420 K	460 K	620 K	340 K	280 K
Mag (dBm/Hz)	-123.7	-111.4	-121.9	-112.4	-124.6	-124.6

Frequency(MHz)	3786.18	3792.17	4000.03	4205.14	4250.02	4283.41
BW (Hz)	160 K	300 K	540 K	200 K	360 K	260 K
Mag (dBm/Hz)	-124.3	-122.8	-119.9	-124.6	-116.0	-125.3

Frequency(MHz)	4583.26	4752.00	4934.95	4951.31	4986.97	4999.92
BW (Hz)	200 K	240 K	260 K	220 K	180 K	200 K
Mag (dBm/Hz)	-125.2	-124.5	-125.0	-125.1	-125.0	-114.1

Table 4-2 Position, BW and Magnitude of the existing signals between 3 to5GHz

All these characteristics provide us very open space to explore the usability of the cognitive radio (CR) strategy. But at the same time, how to avoid sending our special CR group signal on top of the signals that have been there already, which have higher priority and are called Primary Users (PU)? And given the existence of these either big or relative small amplitude PU signals, how do we cancel or mitigate them so that they won't saturate our front-end processing devices, such as amplifiers and Analog to Digital Converter? In the following section we are going to first focus on providing a proposed method to cancel the interference and more research is going to be done as the future work on the Cognitive Radio project which will be described in the next chapter.

4.5 Proposed Cancellation Method

4.5.1 System Concept

Seeing from the diagram Fig 4.15 below, if we can assume the loop is fast enough, which satisfy the assumption that the interference is relative stable during this short period of time, we can try to force the difference between the raw estimation of the interference I_r and the real interference almost the same, by using the adaptive filter.

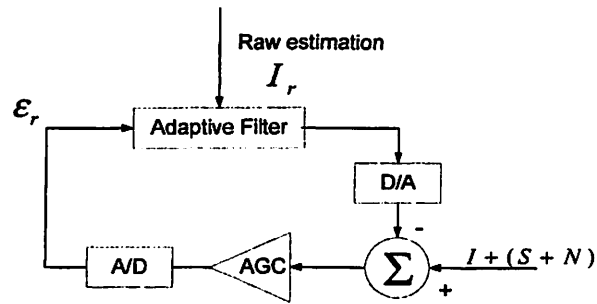


Figure 4.17 Concept of the interference cancellation system

I denotes the real interference, S is our CR group signal, which has been selected by the transmitter to avoid interfering the PUs, N is the noise both from the hardware (thermal noise and distortion generated by the devices) and the channel (some colored noise which are correlated to each other), and ϵ_r is the digital code representing, ideally, the CR signal, if the AGC is a good LNA and the adaptive filter works great to estimate the interference signal.

However, too many ideal assumptions are made here and we need to take care of the reliability of them. Assumptions are addressed as follows:

- *Loop is fast enough to avoid time-varying interference problem*
- *Interference is much larger than the signal we trying to send*
- *INR (interference to noise ratio) is high for all channels*

Although above constraints sound incomplete, the real existing interference are within the circle defined by these assumptions.

Though the real implementation will be much more complicated than the outline we wrote here, this can work as an initial start, and several other considerations are made below.

The difference between the receiver input and the adaptive filter output is calculated in analog domain instead of the digital domain. The reason for this choice is because it is easier to compare and generate differences between two

signals. Depends on the supply voltage and the range of raw estimated and real interference signals, thus the required Common Mode Rejection Ratio (CMRR), Bandwidth and gain specification, it is easy to design an two-input differential amplifier that senses the difference of the interference in analog domain and controls the gain of the AGC.

Startup problem is easy to be solved. Given a good AGC, which has moderate gain variation range, the initial value/waveform of the raw estimated interference is not so important, since only after several cycles the close loop above will be able to adjust the estimation within a very small error step to the real interference.

AGC is used to amplify the interference/error signal all the way up to fulfill ADC's scale. Not only the error of the real and estimated interference performs as the control signal of AGC, but also we want to take full advantage of our dedicated ADC by using the output of the ADC to control the AGC at the same time. By this means, we improve the sensitivity of our system, however with the two control signal being one analog one digital may bring some hidden trouble.

4.5.2 Signal Protection

In order for the adaptive filter to filter out the interference, there has to be some way to protect the signal. Two possible ways are proposed here, coming with the basic idea of creating a subspace by adding constraints at the transmitter side when sending out signal.

- *Coding constraint*

Similar to multi-user detection, while coding, we make our signal orthogonal

to the interference, so it becomes separable for the adaptive filter, thus protecting it from filtering.

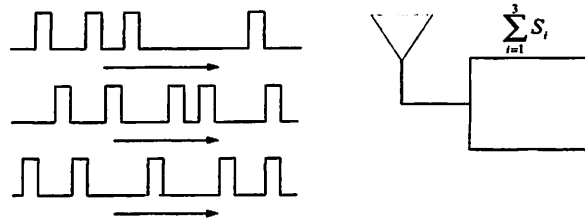


Figure 4.18 Multi-user Detection

- *Passband constraint*

The constraint is defined by creating a non-filterable band for the adaptive filter. There will be some case that some big interference also exists in this certain band, for this case, the transmitter will be asked to make a decision to avoid transmission in this band. Thus, we reserve a dynamic band while satisfying the requirement of not interrupting other PUs which lie within this reservation interval.

It can be seen here that we are trying to solve the interference problem by twisting both the transmitter and receiver sides, so that we can still keep our receiver complexity within the ability we can solve and push some requirement to the transmitted signal to simplify receiver design.

4.5.3 Up convert/Down Convert

We also have the choice to deal with this interference cancellation problem either up convert the baseband as shown in Fig 4.17 or down convert the signal like Fig 4.18. Compare these two options, conclusion will be given as doing up converting is more preferable, and the main reason is from the point of view of getting rid of the distortion problem, but definitely it consumes more power by going this way.

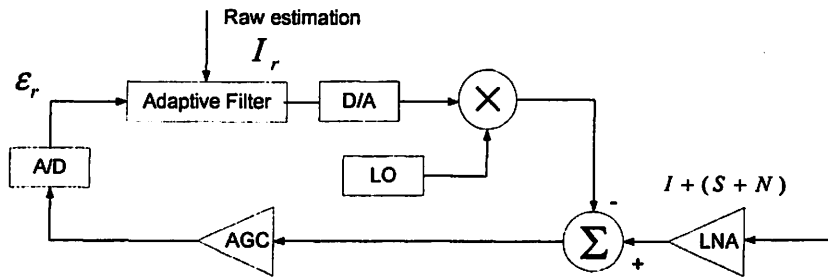


Figure 4.19 Up convert to do cancellation

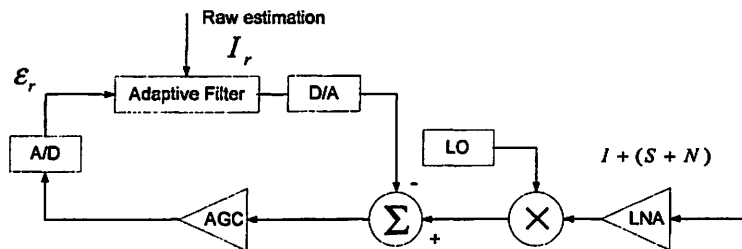


Figure 4.20 Down convert signal to do cancellation

If we do down converting, going along the loop, the distortion generated from the low noise amplifier (LNA) and local oscillator (LO) will appear all the way, since the rest of the loop will treat them as signal. On the other side, if up converting is performed, the 2nd order distortions will be imported like interference, thus easy to be filtered. But 3th order harmonics will stay in the loop, since we will not be able to distinguish whether they are signal or not.

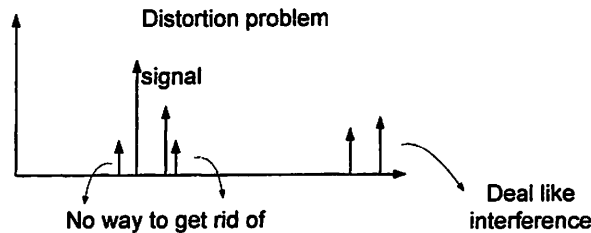


Figure 4.21 Distortion causes unwanted interference in signal band

Commonly saying, it is always not a good choice to process the signals in high frequency. There will be a lot of disadvantages, such as power consuming problem, compared with baseband. Balancing point needs to be find before drawing the final conclusion and it will be done in the future work.

Chapter5 Future Work and Conclusions

As we claimed at the very beginning, the importance of knowing the property of the channel is paramount to that of making an effective design. This work presents the basic idea of spatial channel characterization for both Ultra Wide Band signal and the existing interference. Multi-Antenna system is used and benefits of diversity from different array shapes (short/long, linear/cross /rectangular) are shown by comparing the radiation pattern like beamwidth and sidelobe. The interconnection between the antennas inherently provides us more information, such that we can improve the capacity of the wireless communication system.

On achieving our goal of learning how the UWB signal and interference (PUs) are working, we first carefully setup the system by selecting appropriate equipment and fighting with the thermal noise and quantization noise, which are the two main measurement noise sources in the system by which we make the input referred noise as low as possible to avoid potential for deterioration of the reliability of the received data.

With good noise immunity, characterization of sending UWB signal and pure interference are done in time domain. Clean map of interferences from 0 to 6GHz are given by choosing a reasonable power spectrum density estimation, during which a subtle deconvolution method, called the multi-template subtractive deconvolution method is used. Interferences' locations as well as magnitudes will lead us to the system design view of specification, such as dynamic range (DR) and sensitivity for the future work. What the clean map shows tell us a very important fact that the current spectrum occupancy is far

from crowded, rather, it is regionally empty within a large frequency range. Percentage, position and bandwidth for interferences measured in Berkeley Downtown are given as reference. Moreover, at last, a proposal for interference cancellation is brought out, although it is not complete in terms of only discussing the ideal cases, but it will be of great interest to the future research work when we need to design the cognitive radio system which will smartly adapt itself to live in the aperture of the real interference-existing system.

It is going to be a challenging task to make the first version of implementable cognitive radio system. Not too many real implementation issues are agreed on yet. However, seeing that the flexibility of both the system level and physical level will be greatly improved with much better spectrum efficiency, there is reason one can believe cognitive radio is going to bring a revolution to the world of wireless communication system. And we will be ready to take this challenge.

Appendix A: Program for Receiver Side

```
//Standard include files
#include <stdio.h>           /* location of: printf() */
#include <stdlib.h>         /* location of: atof(), atoi() */
#include <windows.h>
#include <string.h>

//Self defined include files
#include "gpibdecl.h"      /* prototypes, global declarations, constants */
#include "wsc.h"

//Function prototypes
//For telescope
void tel_init( void );      /*initialize the telescope*/
void tel_setAz( int );     /*Set the azimuth of the telescope*/
void tel_setEl( int );     /*Set the elevation of the telescope*/ //<— I modified
void tel_move( void );     /*Move the telescope*/

//For oscilloscope
void initialize( void );   /* initialize the scope */
void acquire_data( int ); /* digitize signals */
void transfer_data( int, /* transfers waveform data from scope to PC */
                  int, int, int); //<—
int convert_data( int, int ); /* converts data to time/voltage values */
void store_csv( FILE *, int ); /* stores voltages to variable file format */

void store_csv1( FILE *, int ); /* stores time values to variable file format*/

//General
void Prompt_filename(void); /*prompt user for filenames of the 4
channels*/
void AzEl_measurement(int, int, int, int);
void ElAz_measurement(int, int, int, int);
void AzOnly_measurement(int, int, int); //
void ElOnly_measurement(int, int, int, int);
void Single_measurement(int, int, int); //

//Global constants
//FOR TELESCOPE
```

```

#define QUOTE 0x22                /*ASCII code for "*/
#define Port 0                    /*COM port = 1*/
#define Baud 9600                 /*Baud rate = 9600*/
#define Parity WSC_NoParity      /*No parity*/
#define StopBits WSC_TwoStopBits /*Two stop bits*/
#define DataBits WSC_WordLength8 /*8 data bits*/
#define CharWait (long)10        /*10ms delay between characters*/

static char Temp[512];           /*buffer holding the return message*/

//Global variables
//FOR OSCILLOSCOPE
int count;
double xorg,xinc;                /* values necessary for conversion of data
*/
double yorg,yinc;
int Acquired_length;
//char data[MAX_LENGTH];        /* data buffer */
short data[MAX_LENGTH];
short garbage[MAX_LENGTH];
double time_value[MAX_LENGTH];  /* time value of data */
double volts[MAX_LENGTH];       /* voltage value of data */

char filename[4][30];           /*hold the files name for the 4 channels*/

int main()
{

int Az_angle_inc;               /*azimuth angle to turn in each step*/
int El_angle_inc;
int Az_angle;                   /*azimuth angle to go*/
int El_angle;                   /*elevation angle to go, can be negative this
time*/ //<---I modified
int RESPONSE_WAIT;             /*Time to wait for each turning in second*/
int choice;                     /*Store the choice of the user*/

#ifdef WITHSCOPE
if( init_IO() ) {
#endif
while (1) {                     /* initialize interface and device sessions */
/* note: routine found in sicl_IO.c or
natl_IO.c */

```

```

    initialize();                /* initialize the scope interface and set
up SRQ */

    tel_init();                  /* initialize the telescope*/

    printf("1. Az/El  measurement\n");
    printf("2. El/Az  measurement\n");
    printf("3. Az only measurement\n");
    printf("4. El only measurement\n");
    printf("5. Single measurement\n");
    printf("6. Quit\n");
    printf("Enter your choice:");
    scanf("%d",&choice);
    if (choice == 6)
        break;
#ifdef WITHSCOPE
    Prompt_filename();
#endif
    switch(choice) {
    case 1: /*Az/El measurement*/
        printf("Enter El angle:");
        scanf("%d",&El_angle);          //notice that user can enter negative
number this time
        printf("Enter Az increment:");
        scanf("%d",&Az_angle_inc);
        printf("Enter El increment:");
        scanf("%d",&El_angle_inc);
        printf("Enter delay between turning:");
        scanf("%d",&RESPONSE_WAIT);

        AzEl_measurement(Az_angle_inc,El_angle_inc,RESPONSE_WAIT,El_angle);

        break;

    case 2: /*El/Az measurement*/
        printf("Enter El angle:");
        scanf("%d",&El_angle);          //notice that user can enter negative
//number this time
        printf("Enter Az increment:");
        scanf("%d",&Az_angle_inc);
        printf("Enter El increment:");
        scanf("%d",&El_angle_inc);
        printf("Enter delay between turning:");

```

```

scanf("%d",&RESPONSE_WAIT);

ElAz_measurement(Az_angle_inc,El_angle_inc,RESPONSE_WAIT,El_angle);

break;

case 3: /*azimuth plane at a particular elevation*/
printf("Enter El angle:");
scanf("%d",&El_angle);          //notice that user can enter negative
//number this time
printf("Enter Az increment:");
scanf("%d",&Az_angle_inc);
printf("Enter delay between turning:");
scanf("%d",&RESPONSE_WAIT);

AzOnly_measurement(El_angle,Az_angle_inc,RESPONSE_WAIT);

break;

case 4: /*elevation plane at a particular azimuth angle*/
printf("Enter El angle:");
scanf("%d",&El_angle);
printf("Enter Az angle:");
scanf("%d",&Az_angle);
printf("Enter El increment:");
scanf("%d",&El_angle_inc);
printf("Enter delay between turning:");
scanf("%d",&RESPONSE_WAIT);

ElOnly_measurement(Az_angle,El_angle_inc,RESPONSE_WAIT,El_angle);

break;

case 5: /*a single coordinate*/
printf("Enter Az angle:");
scanf("%d",&Az_angle);
printf("Enter El angle:");      //notice that user can enter negative
number this time
scanf("%d",&El_angle);
printf("Enter delay between turning:");
scanf("%d",&RESPONSE_WAIT);

Single_measurement(Az_angle,El_angle,RESPONSE_WAIT);

```

```

        break;
    }

    SioDone(Port);          /*Close the COM port*/

}

#ifdef WITHSCOPE

    close_IO( );          /* close interface and device sessions */
                          /* note: routine found in siel_IO.c */
}
#endif
return 0;
}

void Prompt_filename(void)
{
    int i;
    // FILE *fp;

    for(i=0;i<4;i++){
        printf("Enter filename for channel %d (less than 20 characters):",i+1);
        scanf("%s",filename[i]);
        // fp = fopen(filename[i], "ab");
        // fprintf(fp, "ch%d=[",i+1);
        // fclose(fp);
    }
}

void AzEl_measurement(int Az_angle_inc,int El_angle_inc,int RESPONSE_WAIT,int
El_angle)
{
    int az;                /*azimuth angle to go to*/
    int el;                /*elevation angle to go to*/
    int channel;
    int NumByteRead;      /*number of bytes returned by telescope*/
    int tx=-35;

    for(az=0; az<= 360; az+=Az_angle_inc){ //deal with :SzDDD*MM#, setting azimuth
angle

        tel_setAz(az);

```

```

        for(el=El_angle; el<=90; el+=El_angle_inc) { //deal with :Sa+DD*MM#,
setting elevation angle

        tel_setEl(el);

        tel_move(); //deal with the :MS#

        printf("Turning...");
        Sleep(RESPONSE_WAIT*1000); //Wait until the telescope finishes turning

        NumByteRead = SioGets(Port, (LPSTR)Temp, 512);
        printf("Serial device returning %d bytes\n", NumByteRead);
        Temp[NumByteRead] = '\0';
        printf("%s\n", Temp);

        //Add codes here to save the waveform
        for(channel=1; channel <=4; channel+=1) {
                printf("Acquiring
channel=%d, az=%d, el=%d, tx=%d\n", channel, az, el, tx);
                #ifdef WITHSCOPE
                        acquire_data(channel); // capture the data */
                        transfer_data(channel, az, el, tx); // transfer waveform
//data to the PC from scope */
                #endif
                }
        }

        tel_setEl(0); //Return the telescope back to 0 elevation*/
        tel_move();
        printf("Returning to 0 elevation...");
        Sleep(RESPONSE_WAIT*4000); //Wait until the telescope return to 0
elevation*/
    }

}

void ELAz_measurement(int Az_angle_inc, int El_angle_inc, int RESPONSE_WAIT, int
El_angle)
{

    int az; //azimuth angle to go to*/
    int el; //elevation angle to go to*/
    int channel;

```

```

int NumByteRead;          /*number of bytes returned by telescope*/
int tx=-35;

for(el=El_angle; el<=90; el+=El_angle_inc) { //deal with :Sa+DD*MM#, setting
//elevation angle

    tel_setEl(el);

    for(az=0; az<= 360; az+=Az_angle_inc) { //deal with :SzDDD*MM#, setting
//azimuth angle
        tel_setAz(az);

        tel_move();    //deal with the :MS#

        printf("Turning...");
        Sleep(RESPONSE_WAIT*1000); //Wait until the telescope finishes turning

        NumByteRead = SioGets(Port, (LPSTR)Temp, 512);
        printf("Serial device returning %d bytes\n", NumByteRead);
        Temp[NumByteRead] = '\0';
        printf("%s\n", Temp);

        //Add codes here to save the waveform
        for(channel=1; channel <=4; channel+=1) {
            printf("Acquiring
channel=%d, az=%d, el=%d, tx=%d\n", channel, az, el, tx);
            #ifdef WITHSCOPE
                acquire_data(channel);    /* capture the data */
                transfer_data(channel, az, el, tx);    /* transfer waveform
//data to the PC from scope */
            #endif
        }

    }

    tel_setAz(360);
    tel_move();
    Sleep(RESPONSE_WAIT*1000);

    az=az-Az_angle_inc;

```



```

    for(az=-120; az>=0; az-=120){
        tel_setAz(az);
        tel_move();
        printf("Turning...\n");
        Sleep(50000); //Wait until the telescope finishes turning
    }

    //Cancel the offset after turning 360, and make the AutoStar show 0

    tel_setAz(0);
    tel_move();
    printf("Turning...and set az=0\n");
    Sleep(RESPONSE_WAIT*1000);

    printf("Returning to 0 TODO...\n");
    //Sleep(RESPONSE_WAIT*10000);
    //tel_setAz (5); /*Return the telescope back to 0 azimuth*/
    //tel_move();
    //Sleep(RESPONSE_WAIT*1000);    /*Wait until the telescope return to 0
azimuth*/
    }

}

void AzOnly_measurement(int El_angle,int Az_angle_inc,int RESPONSE_WAIT)
{

    int az;
    int channel;
    int NumByteRead;          /*number of bytes returned by telescope*/
    int tx=0;
    // FILE *fp;

    tel_setAz(0);
    if (El_angle!=0) {
        tel_setEl(El_angle);          //notice that this time we can go negative
        tel_move();
        Sleep(RESPONSE_WAIT*4000);
    }

    NumByteRead = SioGets(Port, (LPSTR)Temp, 512);

```

```

printf("Serial device returning %d bytes\n", NumByteRead);
Temp[NumByteRead] = '\0';
printf("%s\n", Temp);

for(az=0; az<= 360; az+=Az_angle_inc) { //deal with :SzDDD*MM#, setting azimuth
//angle

    tel_setAz(az);
    tel_move();

    printf("Turning...");
    Sleep(RESPONSE_WAIT*1000); //Wait until the telescope finishes turning

    NumByteRead = SioGets(Port, (LPSTR)Temp, 512);
    printf("Serial device returning %d bytes\n", NumByteRead);
    Temp[NumByteRead] = '\0';
    printf("%s\n", Temp);

    //Add codes here to save the waveform
    for(channel=1; channel <=4; channel+=1){
        printf("Acquiring,
channel=%d, az=%d, el=%d, tx=%d\n", channel, az, El_angle, tx);
#ifdef WITHSCOPE
            acquire_data(channel);        /* capture the data */
            transfer_data(channel, az, El_angle, tx);        /* transfer waveform
//data to the PC from scope */
#endif
        }

    }

    tel_setAz(360);
    tel_move();
    Sleep(RESPONSE_WAIT*1000);

    for(az-=120; az>=0; az-=120){
        tel_setAz(az);
        tel_move();
        printf("Turning...");
        Sleep(60000); //Wait until the telescope finishes turning
    }

// for(channel=1; channel<=4; channel++){

```

```

//      fp = fopen(filename[channel-1], "ab");
//      fprintf(fp, "];%c%c", (char)13, (char)10);
//      fclose(fp);
//  }
}

void ElOnly_measurement(int Az_angle, int El_angle_inc, int RESPONSE_WAIT, int
El_angle)
{
    int el;
    int channel;
    int NumByteRead;
    //FILE *fp; /*number of bytes returned by telescope*/

//To get ground bound
    int tx; //<-----

    tel_setAz(Az_angle);
    tel_setEl(El_angle);
    tel_move();
    Sleep(RESPONSE_WAIT*3000);

    NumByteRead = SioGets(Port, (LPSTR)Temp, 512);
    printf("Serial device returning %d bytes\n", NumByteRead);
    Temp[NumByteRead] = '\0';
    printf("%s\n", Temp);

    for(tx=0;tx<=3;tx+=1) { //<-----

        for(el=El_angle; el<= 0; el+=El_angle_inc){ //deal with :SzDDD*MM#,
//setting azimuth angle

            tel_setEl(el);
            tel_move();

            printf("Turning...");
            Sleep(RESPONSE_WAIT*1000); //Wait until the telescope finishes turning

            NumByteRead = SioGets(Port, (LPSTR)Temp, 512);
            printf("Serial device returning %d bytes\n", NumByteRead);
            Temp[NumByteRead] = '\0';
            printf("%s\n", Temp);
        }
    }
}

```

```

        //Add codes here to save the waveform
        for(channel=1; channel <=4; channel+=1){
            printf("Acquiring,
channel=%d, az=%d, el=%d, tx=%d\n", channel, Az_angle, el, tx); //<-----
            #ifdef WITHSCOPE
                acquire_data(channel);      /* capture the data */
                transfer_data(channel, Az_angle, el, tx); //<-----      /*
//transfer waveform data to the PC from scope */
            #endif
        }

        tel_setEl(EI_angle);                /*Return the telescope back to 0
elevation*/
        tel_move();
        printf("Returning to max_negative elevation...");
        Sleep(RESPONSE_WAIT*4000);        /*Wait until the telescope return to 0
elevation*/
    } //<-----
}

```

```

void Single_measurement(int Az_angle, int El_angle, int RESPONSE_WAIT)
{
    int az;
    int channel;
    int NumByteRead;          /*number of bytes returned by telescope*/
    int tx=-35;
    az=Az_angle;

    tel_setAz(az);
    tel_setEl(El_angle);
    tel_move();

    printf("Turning...");
    Sleep(RESPONSE_WAIT*3000);

    NumByteRead = SioGets(Port, (LPSTR)Temp, 512);
    printf("Serial device returning %d bytes\n", NumByteRead);
    Temp[NumByteRead] = '\0';
    printf("%s\n", Temp);

    for(channel=1; channel <=4; channel+=1){

```

```

        printf("Acquiring,
channel=%d, az=%d, el=%d, tx=%d\n", channel, Az_angle, El_angle, tx);
        #ifdef WITHSCOPE
            acquire_data(channel);        /* capture the data */
            transfer_data(channel, Az_angle, El_angle, tx);        /* transfer
waveform data to the PC from scope */
        #endif
    }
}

```

```

/*
* Function name: tel_init
* Parameters: none
* Return value: none
* Description : This routine initializes the telescope
                configuring COM port 1 with parameters
                specified at the very beginning of the program
*/

```

```

void tel_init( void )
{
    /* reset (initialize) the port */
    SioParms(-1, Parity, StopBits, DataBits);
    SioReset(Port, 1024, 1024);
    SioBaud(Port, Baud);
    SioDTR(Port, 'S');
    SioRTS(Port, 'S');

    printf("    Port : COM%1d\n", 1+Port);
    printf("    Baud : %d\n", Baud);
    printf("    DTR : set\n");
    printf("    RTS : set\n");
    printf("    Parity : %d\n", Parity);
    printf("CharWait : %ld\n", CharWait);
    printf("StopBits : %d\n", StopBits);
    printf("DataBits : %d\n", DataBits);
}

```

```

/*
* Function name: tel_setAz
* Parameters: azimuth angle
* Return value: none
* Description: This routine sets the azimuth angle that the telescope should move

```

```

*/
void tel_setAz(int k)
{
    SioPutc(Port, ':'); Sleep(CharWait);
    SioPutc(Port, 'S'); Sleep(CharWait);
    SioPutc(Port, 'z'); Sleep(CharWait);

    if(k<=9) {
        SioPutc(Port, '0'); Sleep(CharWait);
        SioPutc(Port, '0'); Sleep(CharWait);
        SioPutc(Port, (char)(k+48));
    }
    else if( (k>=10) && (k<=99) ){
        SioPutc(Port, '0'); Sleep(CharWait);
        SioPutc(Port, (char)((k/10)+48)); Sleep(CharWait);
        SioPutc(Port, (char)((k%10)+48)); Sleep(CharWait);
    }
    else{
        SioPutc(Port, (char)((k/100)+48)); Sleep(CharWait);
        SioPutc(Port, (char)(((k%100)/10)+48)); Sleep(CharWait);
        SioPutc(Port, (char)((k%10)+48)); Sleep(CharWait);
    }

    SioPutc(Port, '.'); Sleep(CharWait);
    SioPutc(Port, '0'); Sleep(CharWait);
    SioPutc(Port, '0'); Sleep(CharWait);
    SioPutc(Port, '#'); Sleep(CharWait);
}

/*
* Function name: tel_setEl
* Parameters: elevation angle
* Return value: none
* Description: This routine sets the elevation angle that the telescope should
move
*/
void tel_setEl(int j) //short instead of unsigned short since we want negative this
time
{
    SioPutc(Port, ':'); Sleep(CharWait);
    SioPutc(Port, 'S'); Sleep(CharWait);
    SioPutc(Port, 'a'); Sleep(CharWait);

    if(j<0) {

```

```

        SioPutc(Port, '-'); Sleep(CharWait);
        j = -j;
    }
    else{
        SioPutc(Port, '+'); Sleep(CharWait);
    }

    if(j<=9) {
        SioPutc(Port, '0'); Sleep(CharWait);
        SioPutc(Port, (char)(j+48));
    }
    else{
        SioPutc(Port, (char)((j/10)+48)); Sleep(CharWait);
        SioPutc(Port, (char)((j%10)+48)); Sleep(CharWait);
    }

    SioPutc(Port, '.'); Sleep(CharWait);
    SioPutc(Port, '0'); Sleep(CharWait);
    SioPutc(Port, '0'); Sleep(CharWait);
    SioPutc(Port, '#'); Sleep(CharWait);

}

/*
 * Function name: tel_move
 * Parameters: none
 * Return value: none
 * Description: This routine turns the telescope to the specified azimuth
                and elevation
 */
void tel_move(void)
{
    SioPutc(Port, ':'); Sleep(CharWait);
    SioPutc(Port, 'M'); Sleep(CharWait);
    SioPutc(Port, 'A'); Sleep(CharWait);
    SioPutc(Port, '#'); Sleep(CharWait);
}

/*
 * Function name: initialize
 * Parameters: none
 * Return value: none
 * Description: This routine initializes the oscilloscope for
                proper acquisition of data. The instrument is reset to a

```

```

*           known state and the interface is cleared. System
*           headers are turned off to allow faster throughput and
*           immediate access to the data values requested by queries.
*           The oscilloscope time base, channel, and trigger
*           subsystems are then configured. Finally, the acquisition
*           subsystem is initialized.
*/

```

```

void initialize( void )
{
// FILE *init_file;
// char line[1024];

write_IO("RST"); /* reset scope - initialize to known state */
write_IO("CLS"); /* clear status registers and output queue */

write_IO("SYSTEM:HEADER OFF"); /* turn off system headers */

// init_file = fopen ("parameter", "r");
// if (!init_file) {
//     fprintf (stderr, "Cannot open parameter file. Using defaults.\n");

/* initialize time base parameters to center reference,
* 100 ns full-scale (100 ns/div) */
write_IO("TIMEbase:REFERENCE LEFT;RANGE 30e-9; POSition 13e-9");

/* initialize 0.12 V full-scale (15 mV/div) */
write_IO("CHANnel1:RANGE 0.04;OFFSet +0");
write_IO("CHANnel2:RANGE 0.04;OFFSet +0");
write_IO("CHANnel3:RANGE 0.04;OFFSet +0");
write_IO("CHANnel4:RANGE 0.04;OFFSet +0");

/* initialize trigger info: AUX signal on positive slope at 80 mV */
write_IO("TRIGger:EDGE:SOURce AUX;SLOPe POSitive");
write_IO("TRIGger:SWEep AUTO");
write_IO("TRIGger:LEVel AUX,5");

/* initialize acquisition subsystem */
/* Real time acquisition - no averaging; memory depth 1,000,000 */

```



```

        write_IO(":ACQuire:MODE RTIME;AVERAge ON;AVERAge:COUNT 16");
        write_IO(":ACQuire:POINts 1000");
        write_IO(":ACQuire:SRATe 20E+9;INTERpolate ON");

// } else {
//
//     while (fscanf (init_file, "%s\n", line) > 0) {
//         if (strlen (line) == 0)
//             continue;

//         write_IO (line);
//     }
//     fclose (init_file);
// }

} /* end initialize() */

/*
 * Function name:  acquire_data
 * Parameters:    none
 * Return value:  none
 * Description:   This routine acquires data according to the current
 *                instrument settings.
 */

void acquire_data( int channel )
{
/*
 * The root level :DIGitize command is recommended for acquisition of new
 * data. It will initialize data buffers, acquire new data, and ensure
 * that acquisition criteria are met before acquisition of data is stopped.
 * The captured data is then available for measurements, storage, or
 * transfer to a PC. Note that the display is automatically turned off
 * by the :DIGitize command and must be turned on to view the captured data.
 */

    char command[40];
    char c[2];

    c[0] = (char) (channel+48);
    c[1] = '\0';

    strcpy(command, ":DIGitize CHANnel");

```

```

    strcat(command, c);
    write_IO(command);

    strcpy(command, ":channel");
    strcat(command, c);
    strcat(command, ":DISPlay ON");
    write_IO(command);
/* turn on channel display turned
   off by the :DIGitize command
*/

} /* end acquire_data() */

/*
* Function name: transfer_data
* Parameters:    none
* Return value:  none
* Description:   This routine transfers the waveform conversion factors
*                and waveform data to the PC.
*/

void transfer_data( int channel, int az, int el, int tx)
{

    char temp;
    int header_length;
    char header_str[8];
    FILE *fp;
    FILE *fp1;
    int time_division=0;

    char xinc_str[32], xorg_str[32];
    char yinc_str[32], yorg_str[32];
    char command[40];
    char c[2];
    char str[30];
    char strm[30];

    int nel;
    int ntx;

    int bytes_read;

    c[0] = (char) (channel+48);
    c[1] = '\0';

```

```

strcpy(command, ":WAVEform:SOURce CHANnel");
strcat(command, c);

write_IO(":WAVEform:FORMat WORD");
write_IO(":WAVEform:BYTeorder LSBFirst");
write_IO(command); /* waveform data source channel */
//write_IO(":WAVEform:FORMat BYTE"); /* setup transfer format */

write_IO(":WAVEform:XINCrement?"); /* request values to allow
interpretation of raw data */
bytes_read = read_IO(xinc_str, 32L);
xinc = atof(xinc_str);

write_IO(":WAVEform:XORigin?");
bytes_read = read_IO(xorg_str, 32L);
xorg = atof(xorg_str);

write_IO(":WAVEform:YINCrement?");
bytes_read = read_IO(yinc_str, 32L);
yinc = atof(yinc_str);

write_IO(":WAVEform:YORigin?");
bytes_read = read_IO(yorg_str, 32L);
yorg = atof(yorg_str);

write_IO(":WAVEform:DATA?"); /* request waveform data */
bytes_read = read_IO(&temp, 1L); /* fine the # character */

while(temp != '#')
    bytes_read = read_IO(&temp, 1L); /* fine the # character */

bytes_read = read_IO(header_str, 1L); /* input byte counter */
header_length = atoi(header_str);

/* read number of points to download */
bytes_read = read_IO(header_str, (long)header_length);
Acquired_length = atoi(header_str); /* number of bytes */

printf("acquired length = %d\n", Acquired_length);

```

```

bytes_read = 0;

if (el<0){
    nel=-el;
    if (tx<0){
        ntx=-tx;
        sprintf(str, "%s_%d_n%d_n%d", filename[channel-1], az, nel, ntx);
        sprintf(strm, "%s_%d_n%d_n%d.m", filename[channel-1], az, nel, ntx);
    }
    else{
        sprintf(str, "%s_%d_n%d_%d", filename[channel-1], az, nel, tx);
        sprintf(strm, "%s_%d_n%d_%d.m", filename[channel-1], az, nel, tx);
    }
}
else{
    if (tx<0){
        ntx=-tx;
        sprintf(str, "%s_%d_%d_n%d", filename[channel-1], az, el, ntx);
        sprintf(strm, "%s_%d_%d_n%d.m", filename[channel-1], az, el, ntx);
    }
    else{
        sprintf(str, "%s_%d_%d_%d", filename[channel-1], az, el, tx);
        sprintf(strm, "%s_%d_%d_%d.m", filename[channel-1], az, el, tx);
    }
}

//fp = fopen(filename[channel-1], "ab"); /* open file in binary mode -
append file
// if already exists */
fp = fopen(strm, "ab");
fpl= fopen("Time.csv", "ab"); /* open file to hold the time values*/

//fprintf(fp, "%d %d", az-5, el);
fprintf(fp, "%s=[", str);

bytes_read += read_IO((char*)data, Acquired_length+1);
time_division = convert_data(time_division, (Acquired_length/2)); /* Convert
data to voltage and time */
if((az==0) && (el==0)) store_csv1(fpl, (Acquired_length/2));
store_csv(fp, (Acquired_length/2));

// fprintf(fp, ":%c%c", (char)13, (char)10);

```

```

    fprintf(fp, "];%c%c", (char)13, (char)10);

    fclose( fp );
    fclose( fpl);
} /* end transfer_data() */

/*
 * Function name: convert_data
 * Parameters: none
 * Return value: none
 * Description: This routine converts the waveform data to time/voltage
 *              information using the values that describe the waveform.
 *              These values are stored in global arrays for use by other
 *              routines.
 */

int convert_data(int time_division, int length)
{
    int i;

    for (i = 0; i < length; i++)
    {
        time_value[i] =(time_division * xinc) + xorg; /* calculate time info */
        volts[i] = (data[i] * yinc) + yorg; /* calculate volt info */
        time_division++;
    }

    return time_division;
} /* end convert_data() */

/*
 * Function name: store_csv
 * Parameters: none
 * Return value: none
 * Description: This routine stores the time and voltage information about the
 *              waveform as
 *              time/voltage pairs in a comma-separated variable file format.
 */

```

```

void store_csv(FILE *fp, int length)
{
    int i;

    if (fp != NULL)
    {

        for (i = 0; i < length; i++)
        {
            fprintf( fp, "%e", volts[i]);    /* write volt to file */
        }

    }
    else
        printf("Unable to open file");

} /* end store_csv() */

void store_csv1(FILE *fp1, int length)
{
    int i;

    if (fp1 != NULL)
    {

        for (i = 0; i < length; i++)
        {
            fprintf( fp1, "%e", time_value[i]);
            fprintf( fp1, "%c%c", (char)13, (char)10);
        }
    }
    else
        printf("Unable to open file");
}

```

Appendix B: Program for Transmitter Side

```
//Standard include files
#include <stdio.h>           /* location of: printf() */
#include <stdlib.h>         /* location of: atof(), atoi() */
#include <windows.h>
#include <string.h>
#include <time.h>

//Self defined include files
#include "wsc.h"           /*Member functions for telescope*/

//Function prototypes
//For telescope
void tel_init( void );     /*initialize the telescope*/
void tel_setAz( int );    /*Set the azimuth of the telescope*/
void tel_move( void );    /*Move the telescope using the MA
command*/

int main()
{
    int Current_TxAz;      /*Current transmitter angle*/
    int TxIdle_delay;     /*Time to remain idle for transmitter
telescope*/
    int StepPlusIdle_delay; /*Time to increment plus idle time*/
    time_t temp1, temp2;
    tel_init();

    printf("TxIdle_delay = ");
    scanf("%d",&TxIdle_delay);
    printf("StepPlusIdle_delay = ");
    scanf("%d",&StepPlusIdle_delay);

    for(Current_TxAz=0; Current_TxAz<=330; Current_TxAz+=30) {

        /*Wait for receiver to turn 360 degrees*/
        (void) time(&temp1);
```

```

Sleep(TxIdle_delay*1000);

(void) time(&temp2);

printf("Sum of t1 and t4 = ", (int) temp2-temp1);

/*Turn to next position, i.e. Current_TxAz +30*/
tel_setAz(Current_TxAz+30);
tel_move();
printf("Transmitter turning to %d\n",Current_TxAz+30);

(void) time(&temp1);

Sleep(StepPlusIdle_delay*1000);

(void) time(&temp2);

printf("t5 = ", (int) temp2-temp1);

}

return 0;
}

/*****
* Function name: tel_init
* Parameters: none
* Return value: none
* Description : This routine initializes the telescope
                 configuring COM port 1 with parameters
                 specified at the very beginning of the program
*****/

void tel_init( void )
{
/* reset (initialize) the port */
SioParms(-1, WSC_NoParity, WSC_TwoStopBits, WSC_WordLength8);
SioReset(COM1, 1024, 1024);
SioBaud(COM1, 9600);
SioDTR(COM1, 'S');

```



```

SioRTS(COM1, 'S');

printf("  Port : COM%d\n", COM1+1);
printf("  Baud : %d\n", 9600);
printf("  DTR : set\n");
printf("  RTS : set\n");
printf("  Parity : %d\n", WSC_NoParity);
printf("CharWait : %ld\n", 10);
printf("StopBits : %d\n", WSC_TwoStopBits);
printf("DataBits : %d\n", WSC_WordLength8);
}

/*****
****
*   Function name: tel_setAz
*   Parameters: azimuth angle
*   Return value: none
*   Description: This routine sets the azimuth angle that the telescope should move
*****/
void tel_setAz(int k)
{
  SioPutc(COM1, ':'); Sleep(10);
  SioPutc(COM1, 'S'); Sleep(10);
  SioPutc(COM1, 'z'); Sleep(10);

  if(k<=9) {
    SioPutc(COM1, '0'); Sleep(10);
    SioPutc(COM1, '0'); Sleep(10);
    SioPutc(COM1, (char)(k+48));
  }
  else if( (k>=10) && (k<=99) ) {
    SioPutc(COM1, '0'); Sleep(10);
    SioPutc(COM1, (char)((k/10)+48)); Sleep(10);
    SioPutc(COM1, (char)((k%10)+48)); Sleep(10);
  }
  else {
    SioPutc(COM1, (char)((k/100)+48)); Sleep(10);
    SioPutc(COM1, (char)((k%100)/10+48)); Sleep(10);
    SioPutc(COM1, (char)((k%10)+48)); Sleep(10);
  }

  SioPutc(COM1, '.'); Sleep(10);
  SioPutc(COM1, '0'); Sleep(10);

```

```
SioPutc(COM1, '0'); Sleep(10);  
SioPutc(COM1, '#'); Sleep(10);  
}
```

```
/******
```

```
*****
```

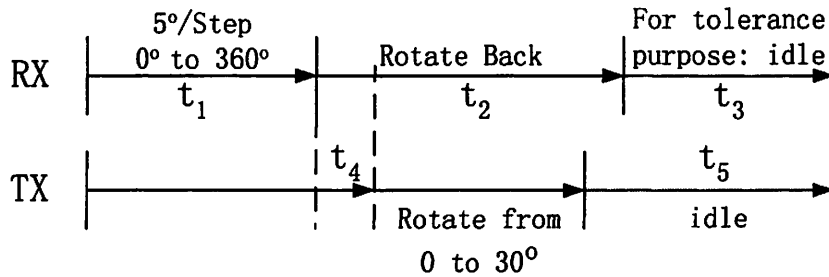
```
* Function name: tel_move  
* Parameters: none  
* Return value: none  
* Description: This routine turns the telescope to the specified azimuth  
and elevation using the MA command
```

```
*****
```

```
*****/
```

```
void tel_move(void)  
{  
SioPutc(COM1, ':'); Sleep(10);  
SioPutc(COM1, 'M'); Sleep(10);  
SioPutc(COM1, 'A'); Sleep(10);  
SioPutc(COM1, '#'); Sleep(10);  
}
```

Appendix C: Timing Diagram of RX & TX



Timing diagrams of Receiver and Transmitter

Measured variables:

$t_1=1440$ seconds

Let $t_3=8$ seconds, $t_4=8$ seconds

$t_2=120$ seconds (expected)

Derived $t_5=120$ seconds

Remarks:

Signal generator frequency and the number of averages will affect t_1 and hence TX_idle_delay variable. This was shown in Table 3.1 already.

Parameters at receiver side:

Parameter at transmitter side:

Step_Delay=15

Tx_Idle_Delay = $t_1+t_4=1448$

RotateBack_Delay= $t_2 = 120$

StepPluseIdle_Delay= $t_5 = 120$

RxIdle_Delay = $t_3 = 8$

Appendix D: Accessing Data Functions

This is a documentation that describes how to use functions, `sized_matrix` and `combine_matrix`, in Matlab to access data, which is collected from a linear-array antenna and a cross-shaped antenna.

Access the data collected from a linear-array antenna:

In Matlab, change the current directory to the folder “data of linear”, then enter the following:

```
>> Channel1_linear;  
>> Channel2_linear;  
>> Channel3_linear;  
>> Channel4_linear;
```

Notice that after the command “Channel1_linear;” is entered, a matrix called `channel1_linear` will be appeared in the workspace. Therefore, after entering the 4 commands above, the following 4 matrices will be appeared in the workspace:

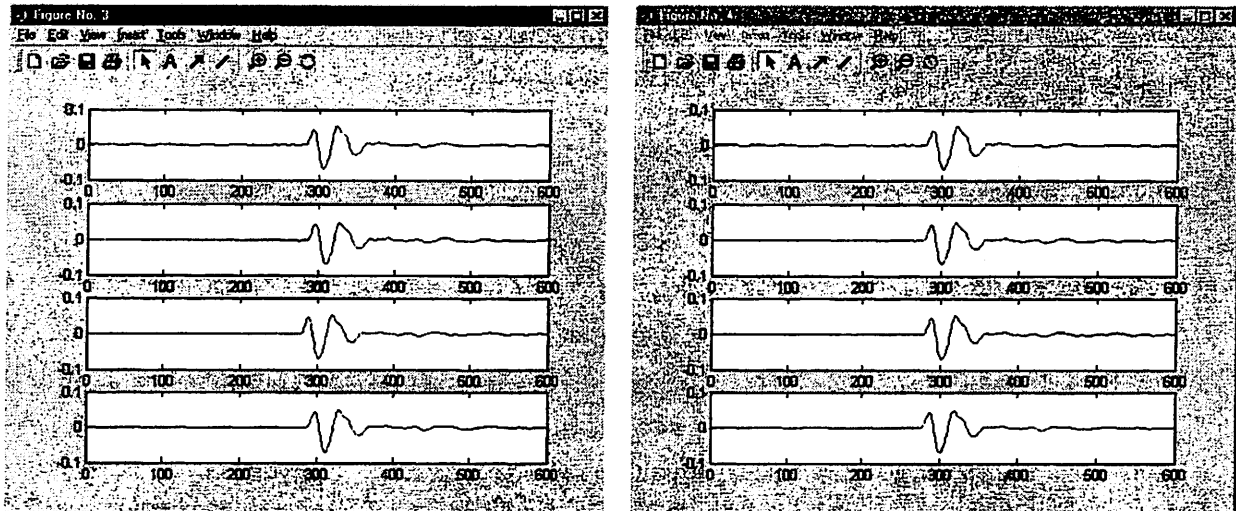
```
Channel1_linear    with size 864x1607  
Channel2_linear    with size 864x1607  
Channel3_linear    with size 864x1607  
Channel4_linear    with size 864x1607
```

However, these 4 matrices do not have the same offset, a function called `sized_matrix` can be used to modified these 4 matrices so that the 4 signals can be lined up and have the same size. Below is an example showing how to use the function `sized_matrix`:

```
>> [c1, c2, c3, c4] = sized_matrix(channel1_linear, channel2_linear,
channel3_linear, channel4_linear, 19.96299, 20.01924, 19.90986, 20.04111);
```

c1, c2, c3, c4 are the new modified matrices that are being lined up. The arguments of the sized_matrix function are the 4 original matrices' names and their offset values correspond to each one of these matrices accordingly. These offset values can be found in the text file called offset_linear.

Original signals received by the antenna at 4 different channels (notice that the peak positions of the signal are different):



New modified matrices at 4 different channels (notice that the peak positions of the signal at each channel are almost the same):

Access the data collected from a cross-shaped antenna:

For the measurement using the cross-shaped antenna, there are 3 sets of data for each channel. These 3 sets of data have different ranges of the transmitter angles.

In Matlab, change the current directory to the folder “data of cross (modified)”, then enter the following:

```
>> channel1_cross_0to60;  
>> channel1_cross_90to330;  
>> channel1_cross_210to330;  
>> channel2_cross_0to60;  
>> channel2_cross_90to330;  
>> channel2_cross_210to330;  
>> channel3_cross_0to60;  
>> channel3_cross_90to330;  
>> channel3_cross_210to330;  
>> channel4_cross_0to60;  
>> channel4_cross_90to330;  
>> channel4_cross_210to330;
```

The command “channel1_cross_0to60;” means that the data is collected from a cross-shaped antenna at channel 1 with transmitter angles from 0 to 60 degrees.

After entering the above commands, 12 matrices will be appeared in the workspace:

channel1_1	called by “channel1_cross_0to60”
channel1_2	called by “channel1_cross_90to330”
channel1_cross3	called by “channel1_cross_210to330”
channel2_1	called by “channel2_cross_0to60”
channel2_2	called by “channel2_cross_90to330”
channel2_cross3	called by “channel2_cross_210to330”
channel3_1	called by “channel3_cross_0to60”
channel3_2	called by “channel3_cross_90to330”
channel3_cross3	called by “channel3_cross_210to330”
channel4_1	called by “channel4_cross_0to60”
channel4_2	called by “channel4_cross_90to330”
channel4_cross3	called by “channel4_cross_210to330”

Notice that for each channel, there are 3 matrices collected at different transmitter angle ranges:

0 to 60 degrees

90 to 330 degrees

210 to 330 degrees

For each of these sets of angles, we would like to line the signal up for the 4 channels using the function `sized_matrix`:

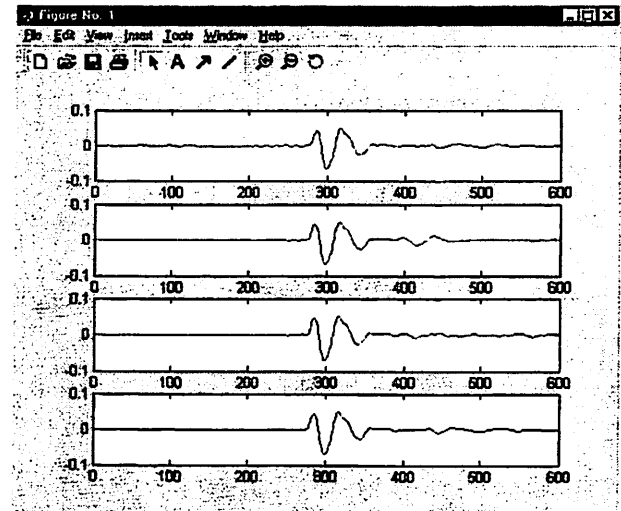
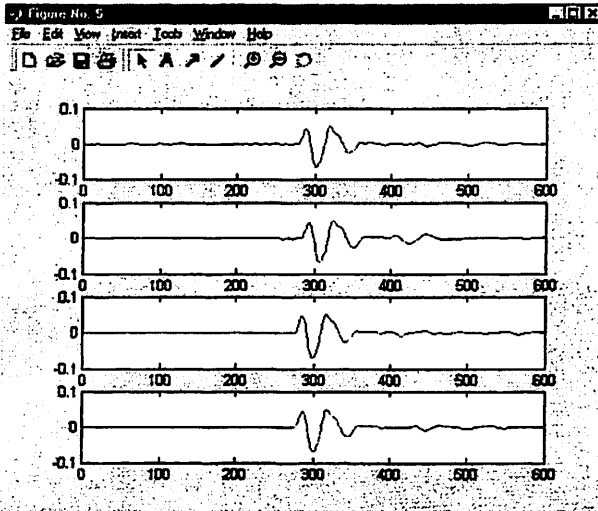
```
>> [c1_0to60, c2_0to60, c3_0to60, c4_0to60] = sized_matrix (channel1_1,  
channel2_1, channel3_1, channel4_1, 19.91298, 19.98486, 19.89021,  
19.90102);
```

```
>> [c1_90to330, c2_90to330, c3_90to330, c4_90to330] = sized_matrix  
(channel1_2, channel2_2, channel3_2, channel4_2, 19.91298, 19.98486,  
19.89021, 19.90102);
```

```
>> [c1_210to330, c2_210to330, c3_210to330, c4_210to330] = sized_matrix  
(channel1_cross3, channel2_cross3, channel3_cross3, channel4_cross3,  
20.11611, 20.14736, 20.05361, 20.10674);
```

The way to use the `sized_matrix` function was introduced in the previous session. The offset values obtained from the cross-shaped antenna can be found in the text file called `offset_cross`.

Original signals received by the antenna at 4 different channels (notice that the peak positions of the signal are different):



New modified matrices at 4 different channels (notice that the peak positions of the signal at each channel are almost the same):

After lining up the signals, the 3 matrices of each channel can then be combined into 1 matrix using the function `combine_matrix`. Below is an example showing how to use this function:

```
>> c1_cross = combine_matrix ( c1_0to60, c1_90to330, c1_210to330);
>> c2_cross = combine_matrix ( c2_0to60, c2_90to330, c2_210to330);
>> c3_cross = combine_matrix ( c3_0to60, c3_90to330, c3_210to330);
>> c4_cross = combine_matrix ( c4_0to60, c4_90to330, c4_210to330);
```

`c1_cross`, `c2_cross`, `c3_cross`, `c4_cross` are the new combined matrices with transmitter angles from 0 to 330 degrees for each of the channel. The arguments of the `combine_matrix` function are the 3 matrices obtained from the same channel but have different ranges of transmitter angles.

References

- [1] <http://cnx.rice.edu/content/m0516/latest/>
- [2] <http://bwrc.eecs.berkeley.edu/Publications/2004/PRESENTATIONS/asypoon.ieee04/ASYPoon.IEEE04.pdf>
- [3] R. B. Ertel, P. Cardieri, K. W. Sowerby, T. S. Rappaport and J. H. Reed, *Overview of Spatial Channel Models for Antenna Array Communication Systems*, IEEE Personal Communications, February 1998
- [4] http://www.techonline.com/community/ed_resource/feature_article/14707
- [5] http://www.cwc oulu.fi/home/oulu_teaser.pdf
- [6] http://web.rsise.anu.edu.au/~mreed/presentations/rod_vtcSlides.pdf
- [7] <http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=37>
- [8] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill Companies, Inc, 2001
- [9] B. Boser, Class notes of EE 247, 2003 Fall
- [10] P. Gray, P. J. Hurst, S.H. Lewis and R.G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th Edition, J. Wiley & Sons, Inc, 2000

- [11] [https:// ewhdbks.mugu.navy.mil/RADIAPAT.HTM](https://ewhdbks.mugu.navy.mil/RADIAPAT.HTM)
- [12] R. C. Hansen, *Phased array antennas*, J. Wiley, 1998
- [13] R. G. Gallager, *Stochastic Process: A Conceptual Approach*
- [14] <http://cnx.rice.edu/content/m12369/latest/>
- [15] R. Murty, *Software-Defined Reconfigurable Radios: Smart, Agile, Cognitive, and Interoperable*, Technology@intel Magazine, pp3, July 2003
- [16] H. Hashemi, Impulse Response Modeling of Indoor Radio Propagation Channels, IEEE Journal on selected areas in communications. Vol 11. No.7, September 1993
- [17] R. G. Vaughan and N. L. Scott, Super-Resolution of Pulsed Multipath Channels for Delay Spread Characterization, IEEE Tran. on Comms., Vol.47, No.3, March 1999
- [18] G. Morrison and M. Fattouche, Super-Resolution Modeling of the Indoor Radio Propagation Channel, IEEE Tran. on Vehicular Technology, Vol.47, No.2, May 1998
- [19] J. Mitola and G. Q. Maguire, Cognitive Radio: Making Software Radios More Personal, IEEE personal Comms, August 1999
- [20] R. J. Cramer, An Evaluation of UWB Propagation Channels, PH. D. thesis, University of Southern California
- [21] <http://www.atnf.csiro.au/projects/ska/WS/wsrf/hfisher1.html>

- [22] D. Tse etc., Proposal for Universal Spectrum Sharing, 2002
- [23] A. Muqaibel, A. Safaai-Jazi, B. Woerner and S. Riad, *UWB Channel Impulse Response characterization Using Deconvolution Techniques*, 45th IEEE International Midwest Symposium on Circuits Systems, August, 2002
- [24] http://www.techonline.com/community/ed_resource/feature_article/14707
- [25] P.M. J. Scheeren, *Interference-reduction techniques for earth stations, and the specific problem of Near-Field site Shielding Against Terrestrial Interference*, IEE Colloquium, Sep. 1988
- [26] Y. Palaskas, Y. Tsvividis, V. Prodanov and V. Boccuzzi, *A "Divide and Conquer" Technique for Implementing Wide Dynamic Range Continuous-Time Filters*, IEEE JSSC, Vol.39, No. 2, February 2004
- [27] <http://bwrc.eecs.berkeley.edu/Seminars/Suh-4.16.04/>, *Novel Printed Ultra-wideband Antenna*, Dr. Seong-Youp Suh
- [28] http://www.aetherwire.com/UWBWG_Archive/attach/UWBWG_Draft_2.0.doc, Aetherwire corp.