Analysis and Implementation of a Novel Single Channel Direction Finding Algorithm on a Software Radio Platform

John Joseph Keaveny

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

MASTER OF SCIENCE
In
Electrical Engineering

Dr. R. Michael Buehrer, Chair
Dr. Jeffrey Reed
Dr. William Tranter

February 11, 2005
Blacksburg, VA

Keywords: Single Channel Direction Finding Systems, Watson-Watt, Pseudo-Doppler, Adcock Antenna Array, Circular Antenna Array, Software-Defined Radio

Analysis and Implementation of a Novel Single Channel Direction Finding Algorithm on a Software Defined Radio Platform

John Joseph Keaveny

Abstract

A radio direction finding (DF) system is an antenna array and a receiver arranged in a combination to determine the azimuth angle of a distant emitter. Basically, all DF systems derive the emitter location from an initial determination of the angle-of-arrival (AOA).

Radio direction finding techniques have classically been based on multiple-antenna systems employing multiple receivers. Classic techniques such as MUSIC [1][2] and ESPRIT use simultaneous phase information from each antenna to estimate the angle-of-arrival of the signal of interest. In many scenarios (e.g., hand-held systems), however, multiple receivers are impractical. Thus, single channel techniques are of interest, particularly in mobile scenarios. Although the amount of existing research for single channel DF is considerably less than for multi-channel direction finding, single channel direction finding techniques have been previously investigated.

Since many of the single channel direction finding techniques are older analog techniques and have been analyzed in previous work, we will investigate a new single channel direction finding technique that takes specific advantage of digital capabilities. Specifically, we propose a phase-based method that uses a bank of Phase-Locked Loops (PLLs) in combination with an eight-element circular array. Our method is similar to the Pseudo-Doppler method in that it samples antennas in a circular array using a commutative switch. In the proposed approach the sampled data is fed to a bank of PLLs which track the phase on each element. The parallel PLLs are implemented in software and their outputs are fed to a signal processing block that estimates the AOA.

This thesis presents the details of the new Phase-Locked Loop (PLL) algorithm and compares its performance to existing single channel DF techniques such as the Watson-Watt and the Pseudo-Doppler techniques. We also describe the implementation of the PLL algorithm on a DRS Signal Solutions, Incorporated (DRS-SS) WJ-8629A Software Definable Receiver with Sunrise ™ Technology and present measured performance results.

For Leslie, without you this would not be possible

Acknowledgements

There are many people that I need to thank because without these people, I wouldn't be here today. First I would like to thank Michael Buehrer, Jeffrey Reed, and William Tranter for serving as my thesis committee. Their technical expertise, advice, and support throughout my career at Virginia Tech have been invaluable.

I would like to thank Dr. R. Michael Buehrer for not only being my advisor but also my good friend. His technical knowledge and mentoring has shaped me into what I am today. During my time at Virginia Tech, he guided me from theoretical concepts, to simulations, to the actual implementation of my thesis work. When I called, he was there; and when I stumbled, he was there to pick me back up. He has been invaluable in my career. Thank you.

I would like to thank DRS Signal Solutions, Incorporated (DRS-SS) for funding my thesis. They provided the software defined radio, WJ-8629A Software Definable Receiver with Sunrise ™ Technology, to MPRG for direction finding algorithm development. Tom Potter and Jeff Silvey were very helpful and went out of their way whenever questions surfaced about the receiver.

I owe a great deal of thanks to all of my mentors who are PhD students. Jody Neel was there to give me a crash course in DSPs and software radios. He was always available to ask questions if I had a problem. Chris Anderson was my RF guru. Not only did we build a homemade anechoic chamber in the MPRG lab, but he also helped me set up a test procedure to validate my algorithm implementation. Swaroop Venkatesh helped me troubleshoot the algorithm with simulations while I was busy trying to implement it on our software radio. He was also a great help when double and triple checking my technical writing. Lastly, Max Robert always helped alleviate stress by listening and offering his services when available.

Nathan Harter and I were good friends when we did our undergraduate work together at St. Louis University. He later followed me out to Virginia Tech to pursue his Masters degree. His help with the antenna switching circuitry and his help in the lab was very beneficial. Nathan will continue exploring our algorithm in more depth for his thesis.

Next I must thank all of my friends that I had made at MPRG. Brian Donlan, Dan Hibbard, Vivek Bharadwaj, Jihad Ibrahim, and Samir Ginde always were there for me and each other. They are a great group of people, and it was a pleasure to discuss our work over buffalo wings.

I need to thank the staff of MPRG. Hilda Reynolds and Jenny Frank were always looking out for us and helping us along the way.

I would like to thank my parents, sister, and family for their support. They were always there for me even though they were 800 miles away.

Lastly, none of this would be possible without the love and support of my lovely wife Leslie. She was always there to listen when days were good and when days were not so good. She may not understand what my work involves, but she does understand when I need her. Without her, none of this would have been possible.

I will always remember and cherish my time at MPRG and Virginia Tech. Go Hokies!

John Keaveny
February 11, 2005

Table of Contents

List of Figures

List of Tables

List of Acronyms

A/D	**Analog-to-Digital Converter**
AOA	**Angle of Arrival**
AM	**Amplitude Modulation**
AWGN	**Additive White Gaussian Noise**
BPSK	**Binary Phase Shift Keying**
CPLD	**Complex Programmable Logic Device**
CW	**Continuous Wave**
DF	**Direction Finding**
DOA	**Direction of Arrival (equivalent to AOA)**
DSP	**Digital Signal Processor**
E	**East**
FFT	**Fast Fourier Transform**
FM	**Frequency Modulation**
LOS	**Line of Sight**
MPRG	**Mobile and Portable Radio Research Group**
N	**North**
PLL	**Phase-Locked Loop**
RF	**Radio Frequency**
Rx	**Receiver**
S	**South**
SNR	**Signal-to-Noise Ratio**
SPDT	**Single Pole Double Throw (RF switch)**
SP4T	**Single Pole Four Throw (RF switch)**
SS	**Switch Signal**
Tx	**Transmitter**
TTL	**Transistor-Transistor Logic**
W	**West**

Chapter 1

Introduction

A radio direction finding (DF) system is an antenna array and a receiver arranged in a combination to determine the azimuth angle of a distant emitter. Basically, all DF systems derive the emitter location from an initial determination of the angle-of-arrival (AOA).

Radio direction finding techniques have classically been based on multiple-antenna systems employing multiple receivers. Classic techniques such as MUSIC [1][2] and ESPRIT use simultaneous phase information from each antenna to estimate the angle-of-arrival of the signal of interest. In many scenarios (*e.g.,* hand-held systems), however, multiple receivers are impractical. Thus, single channel techniques are of interest, particularly in mobile scenarios. Although the amount of existing research for single channel DF is considerably less than for multi-channel direction finding, single channel direction finding techniques have been previously investigated.

When considering single channel direction finding systems, we find that there are two distinct types of DF systems. The first type of DF system is the amplitude-based DF system. Amplitude-based systems determine the bearing of the signal (or the AOA) by analyzing the amplitudes of the output voltages from each antenna element. Amplitude DF systems include the Watson-Watt technique using an Adcock antenna array [3].

The second type of DF system is the phase-based DF system. Phase-based systems use three or more antenna elements that are configured in a way so that the relative phases of their output voltages are unique for every wavefront angle-of-arrival. Phase-based DF systems include the Pseudo-Doppler technique with a commutative switch based antenna array [3].

Since both of the above techniques are primarily analog techniques and have been analyzed in previous work, we will investigate a new single channel direction finding technique that takes specific advantage of digital capabilities. Specifically, we propose a phase-based method that uses a bank of Phase-Locked Loops (PLLs) in combination with an eight-element circular array. Our method is similar to the Pseudo-Doppler method in that it samples antennas in a circular array using a commutative switch. In the proposed approach the sampled data is fed to a bank of PLLs which tracks the phase on each element. The parallel PLLs are implemented in software and their outputs are fed to a signal processing block that estimates the AOA.

This thesis presents the details of the new algorithm and compares its performance to existing single channel DF techniques such as the Watson-Watt and the Pseudo-Doppler techniques. We also describe the implementation of the algorithm on a DRS Signal Solutions Incorporated (DRS-SS), WJ-8629A Software Definable Receiver with Sunrise ™ Technology and present measured performance results. Simulations on a signal with 10dB SNR have shown that the Watson-Watt algorithm and the Pseudo-Doppler algorithm have an accuracy that is worse than the proposed technique by approximately an order of magnitude.

The algorithm was implemented on a single-channel DSP-based software radio with a homemade eight-element circular antenna array. The WJ-8629A software defined radio receiver was provided by DRS-SS in order to implement our algorithm. The implementation was tested using a CW signal at ~1.57068 GHz in a low multipath laboratory environment and outdoors. The performance of the prototype is compared to the data provided by the simulations in Matlab.

Implementation results focus on CW measurements in a relatively benign laboratory environment for proof-of-concept testing. This document will show that the basic version of the algorithm can result in a significant computational burden, thus we investigate a low-complexity approach and demonstrate its performance. It will be shown that a significant computational reduction can be achieved with minimal performance penalty.

1.1 Software Introduction

During our research, all of the single-channel direction finding simulations were performed using the MATLAB 6.1 software. After the simulations were completed, the MATLAB code was then ported to hardware for implementation using the C programming language. The initial C programs were written and tested to prove that the algorithms could be implemented on the TI based software radio. After the C programs were tested and compared to their Matlab counterparts, they were then optimized for the Texas Instruments TMS320C67x Digital Signal Processor.

1.2 Hardware Introduction

1.2.1 DRS Signal Solutions, Incorporated WJ-8629A Software Definable Receiver with Sunrise ™ Technology

The implementation was performed on a Texas Instruments DSP-based WJ-8629A software defined radio provided by DRS-SS. It has a frequency range from 20 to 2700 MHz with 10-Hz resolution, receiver filtering with 22 filter slots (200 Hz to 1.23 MHz), and 5 reserved slots for user-downloadable custom filters The main processing unit is the Texas Instruments TMS320C6701 DSP processor with a maximum computational rate of nearly 1GFlops [4]. The radio allows one to develop algorithms for certain signal processing modules in the C programming language or the TMS320C67x assembly language. Other details of the radio are not listed here due to their proprietary nature. Throughout this thesis we will include only those details necessary for proper understanding of the implementation.

1.2.2 MPRG Antenna Array

The antenna baseline is the geometric line of interconnection between antenna elements. Antenna aperture is defined as the plane surface area near the antenna through which most of the radiation flows. The spacing between antenna elements usually determines the aperture of an array, and since we are using circular arrays, the diameter of the entire circular array determines the array aperture [5].

In order to model the antenna array, assuming a single plane wave impinging on the array, the array manifold vector for a uniform circular array can be written as:

$$U(\theta)=\begin{bmatrix} u_1(\theta) \\ u_2(\theta) \\ \vdots \\ u_M(\theta) \end{bmatrix}=\begin{bmatrix} e^{-j2\pi(R/\lambda)\sin(\varphi)\cos(\theta-\eta_1)} \\ e^{-j2\pi(R/\lambda)\sin(\varphi)\cos(\theta-\eta_2)} \\ \vdots \\ e^{-j2\pi(R/\lambda)\sin(\varphi)\cos(\theta-\eta_M)} \end{bmatrix} \tag{1.1}$$

where R is the radius of the circular antenna array, φ is the elevation angle, θ is the angle of arrival (AOA) of the incoming plane wave, η_m is the angle of the mth antenna element in the azimuthal plane, and λ is the wavelength of the center frequency of interest [5]. For simplicity, the elevation angle is set to 90° in order to consider azimuth angles only. We do not consider the effects of different elevations in this study.

The MPRG antenna array as seen in Figure 1.1 is an eight-element antenna array with a diameter of ~19.1 cm. We desire to have a waveform that completes one wavelength over the diameter of the array which will be discussed in detail in later chapters. Therefore, the frequency of the CW is defined as $f = c/\lambda$ or 1.57068 GHz.

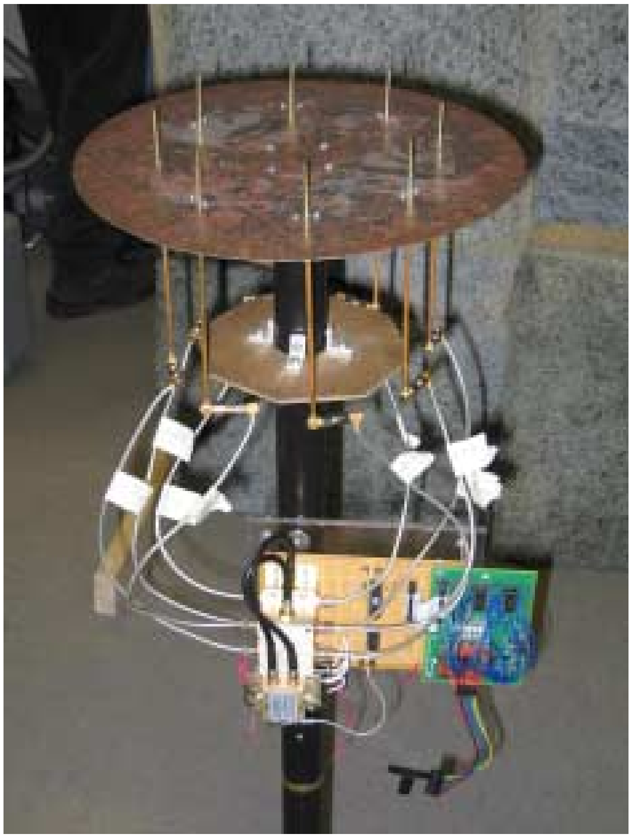


Figure 1.1 MPRG Eight Element Circular Antenna Array

1.3 Thesis description

In Chapter 2, we will discuss the theory behind a couple of known single-channel direction finding algorithms. The algorithms that will be discussed are the amplitude based Watson-Watt algorithm using an Adcock array, and the phase-based Pseudo-Doppler technique.

Chapter 3 will present our novel Phase-locked loop algorithm. Simulations will demonstrate our algorithms improved angular resolution and accuracy. After we discuss the PLL algorithm, Chapter 4 will include simulations of the previous DF algorithms discussed in Chapter 2. These simulations will include the effects of multipath environments, Rayleigh fading channels, and signal-to-noise ratio (SNR).

Chapter 5 will describe the implementation of the PLL algorithm on the WJ-8629A software defined radio discussed above. The WJ-8629A software radio, software architecture, and commutative switching circuit will be discussed in detail. The later part of Chapter 5 will present results from the implementation

The last chapter will present our conclusions on the PLL algorithm. We will discuss the algorithms performance versus the other DF algorithms, its performance during the implementation, lessons learned and future work.

Chapter 2

Introduction to Single Channel Direction Finding

To date, the two primary methods that have been examined for single channel direction finding are the Watson-Watt Method using an Adcock antenna array, and the Pseudo-Doppler Method using a commutative switch with a circular antenna array [6][7]. While little is available in the open literature concerning these two techniques, what is available assumes an analog receiver and operates at relatively low frequencies. Specifically, the Adcock/Watson-Watt algorithm is typically used for frequencies up to about 1000 MHz, while the Pseudo-Doppler algorithm typically has an operational bandwidth from 2–2000 MHz. In this chapter, we will discuss the amplitude-based Watson-Watt technique, the phase-based Pseudo-Doppler method, and an amplitude-based Pseudo-Doppler technique developed as part of the current research. We will discuss their strengths and short-comings and motivate the investigation of new techniques.

2.1 The Watson-Watt Method

Watson-Watt DF is an amplitude-based method that uses the relative amplitude of the output of two antenna arrays arranged according to the Adcock design. The Adcock design consists of four antenna elements in a perpendicular, crossed-baseline configuration as seen in Figure 2.1.

This method can be used for frequencies up to about 1000 MHz. One Adcock pair contains two antenna arrays (four antenna elements) in a perpendicular configuration, with element spacing of less than one half the wavelength at the highest operating frequency. The azimuth gain pattern from each antenna array is obtained by a vector difference of signals from each of two antennas.

The signals seen on the four antennas (labeled North - N, South - S, East - E, and West - W) in complex baseband notation are:

$$
\begin{aligned}
r_n(t) &= m(t)e^{j\frac{2\pi R}{\lambda}\sin(\Phi)} \\
r_s(t) &= m(t)e^{-j\frac{2\pi R}{\lambda}\sin(\Phi)} \\
r_e(t) &= m(t)e^{j\frac{2\pi R}{\lambda}\cos(\Phi)} \\
r_w(t) &= m(t)e^{-j\frac{2\pi R}{\lambda}\cos(\Phi)}
\end{aligned}
\tag{2.1}
$$

where *r(t)* is the received signal, *R* is the radius of the circular antenna array, λ is the wavelength of the center frequency of interest, *m(t)* is a linearly modulated message signal and Φ is the AOA [6]. The East antenna represents our 0^o reference.

The N and S antenna pair creates the Y-axis voltage, which has a maximum gain along the Y-axis. In other words when Φ = 90^o, the east and west signals are equal and thus $x(t) = r_e(t)\text{-}r_w(t) = 0$, whereas $y(t) = r_n(t)\text{-}r_s(t) = 2m(t)$. The E and W antenna pair creates the X-axis voltage, which has maximum gain along the X-axis. In other words when Φ = 0^o, the north and south signals are equal and thus $x(t) = r_e(t)\text{-}r_w(t) = 2m(t)$, whereas $y(t) = r_n(t)\text{-}r_s(t) = 0$.

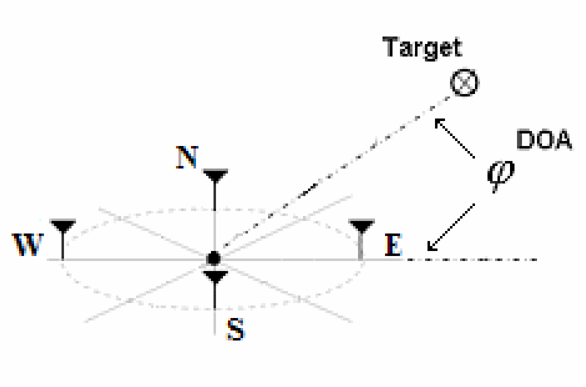


Figure 2.1 Adcock Antenna Array used for Watson-Watt Algorithm

In order to pass the AOA data to the single receiver, each of the X and Y axis voltages have to be combined into a composite signal. In our example in Chapter 4, the two signals are linearly combined to form an AM signal with dual tone modulation in order to pass the data to the single receiver.

After the linearly combined AM signal reaches the receiver and AM demodulation is performed, the estimated AOA ($\hat{\Phi}$) is calculated by taking the arctangent of the N-S difference divided by the E-W difference.

$$\begin{aligned} y(t) &= r_{n-s}(t) = 2j*m(t)\sin\left(\frac{2\pi R}{\lambda}\sin(\Phi)\right) \\ x(t) &= r_{e-w}(t) = 2j*m(t)\sin\left(\frac{2\pi R}{\lambda}\cos(\Phi)\right) \end{aligned} \tag{2.2}$$

$$\begin{aligned} \hat{\Phi}(t) &= \arctan\left(\frac{y(t)}{x(t)}\right) \\ &= \arctan\left(\frac{2j*m(t)\sin\left(\frac{2\pi rR}{\lambda}\sin(\Phi)\right)}{2j*m(t)\sin\left(\frac{2\pi R}{\lambda}\cos(\Phi)\right)}\right) \\ &= \arctan\left(\frac{\sin\left(\frac{2\pi R}{\lambda}\sin(\Phi)\right)}{\sin\left(\frac{2\pi R}{\lambda}\cos(\Phi)\right)}\right) \\ &\approx \arctan\left(\frac{\sin(\Phi)}{\cos(\Phi)}\right) \end{aligned} \tag{2.3}$$

where the approximation holds for small values of $\frac{2\pi R}{\lambda}$, since $\sin(x) \approx x$ for small values of x [6]. If we use the antenna array described in Figure 2.1, we will encounter an 180° phase ambiguity since a negative ratio could correspond to either quadrant 2 or 4 whereas a positive ratio could correspond to either quadrant 1 or 3. If a centrally located omni-directional antenna is included in Figure 2.1, then it can provide basic directional sensing to help eliminate the 180° phase ambiguity [8]. In Chapter 4, we will examine the accuracy of the Watson-Watt algorithm using an Adcock array in various conditions via simulation.

2.2 Pseudo-Doppler Algorithm

The Pseudo-Doppler technique is a phase comparison method that exploits the Doppler shift on successive samples of circularly disposed antenna elements. Measurements of phase differences between the elements of a multi-element antenna array enable the azimuth angle of the arriving signal to be determined. One system of this type is the Pseudo-Doppler method. In principle, an antenna element could be moved in a circular path so that the instantaneous frequency of the received signal would be modified.

Alternatively, a rotating commutative RF switch is used to couple a receiver in rapid sequence to the elements of the array, thereby introducing a frequency shift on the received signal which is extracted by a frequency discriminator. As the antenna moves, it imposes a

Doppler shift on the arriving signal. The magnitude of the Doppler shift is at a maximum as the antenna moves directly toward and away from the direction of the incoming wavefront. There is no apparent frequency shift when the antenna moves orthogonal to the wavefront [9]. The azimuth angle is given by the angular position of the rotor at which zero instantaneous frequency shift occurs. Ambiguity can be removed by taking account of the angles at which maximum positive and negative frequency shifts occur.

As in Figure 2.2, the value of φ_r changes with the sampling position which results in a frequency shift of 0^o when φ_1 is exactly coincident with the incoming signal azimuth angle with an 180^o phase ambiguity. Therefore, near zero frequency shift occurs at angles (45^o) and (-135^o). The ambiguity can be resolved by finding the maximum negative frequency shift at (-45^o) and the maximum positive frequency shift at (135^o).

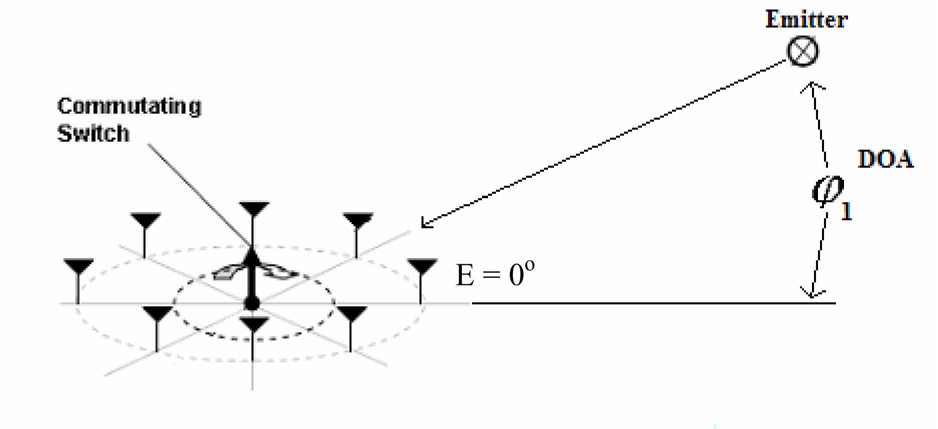


Figure 2.2 Pseudo Doppler Frequency Shift

Consider a linearly modulated signal impinging on an N_a-element circular array

$$x(t) = \mathrm{m}(t)\cos\left(\omega_c \mathrm{t} + \theta\right) \tag{2.4}$$

Assume that the receiver switches from the ith antenna to the $(i + 1)$th antenna every T_s seconds. Now each antenna imposes a phase shift of

$$\frac{2\pi R}{\lambda}\cos\left(\frac{2\pi i}{N_a} - \Phi\right) \tag{2.5}$$

where R is the radius of the circular array, λ is the wavelength of interest, Φ is the angle-of-arrival and $i = 0 \ldots N_a - 1$. Now if the switch changes to the neighboring antenna every T_s seconds, it imposes a time varying phase shift

$$\theta_s(t) = \frac{2\pi R}{\lambda}\cos\left(\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_s)-\Phi\right) \tag{2.6}$$

where *u(t)* is the unit step function. The received signal is then

$$r(t) = m(t)\cos\left(\omega_c t+\theta+\frac{2\pi R}{\lambda}\cos\left(\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_s)-\Phi\right)\right) \tag{2.7}$$

Ignoring for the moment the message signal, the output of an FM discriminator is

$$f(t) = \frac{d}{dt}\left\{\omega_C t+\theta+\frac{2\pi R}{\lambda}\cos\left(\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_S)-\Phi\right)\right\} \tag{2.8}$$

$$f(t) = \omega_C+\frac{d}{dt}\left\{\frac{2\pi R}{\lambda}\cos\left(\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_S)-\Phi\right)\right\} \tag{2.9}$$

$$f(t) = \omega_C-\frac{2\pi R}{\lambda}\sin\left(\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_S)-\Phi\right)\frac{d}{dt}\left\{\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_S)\right\} \tag{2.10}$$

$$f(t) = \omega_C-\frac{2\pi R}{\lambda}\sin\left(\frac{2\pi}{N_a}\sum_{n=1}^{\infty}u(t-nT_S)-\Phi\right)\frac{2\pi}{N_a}\sum_{n=1}^{\infty}\delta(t-nT_S) \tag{2.11}$$

Now, since this is not a true differentiator, but a discrete approximation, there is a delay of $T_s/2$:

$$f(t) = \omega_C-\frac{2\pi R}{\lambda}\frac{2\pi}{N_a}\sum_{n=1}^{\infty}\sin\left(\frac{2\pi}{N_a}\left(n-\frac{1}{2}\right)-\Phi\right)\delta(t-nT_S) \tag{2.12}$$

Now, after down-conversion, we can ignore the carrier term. Thus, we have

$$f(t) = -\frac{2\pi R}{\lambda}\frac{2\pi}{N_a}\sum_{n=0}^{\infty}\sin\left(\frac{2\pi}{N_a}\left(n-\frac{1}{2}\right)-\Phi\right)\delta(t-nT_s) \tag{2.13}$$

The samples for every N_a values can be entered into a vector:

$$f[n]=\begin{bmatrix} -\frac{2\pi R}{\lambda}\frac{2\pi}{N_a}\sin\left(\frac{2\pi}{N_a}\left(-\frac{1}{2}\right)-\Phi\right) \\ -\frac{2\pi R}{\lambda}\frac{2\pi}{N_a}\sin\left(\frac{2\pi}{N_a}\left(\frac{1}{2}\right)-\Phi\right) \\ \cdots \\ -\frac{2\pi R}{\lambda}\frac{2\pi}{N_a}\sin\left(\frac{2\pi}{N_a}\left(N_a-1-\frac{1}{2}\right)-\Phi\right) \end{bmatrix} \tag{2.14}$$

Now, the FFT of this vector is

$$F[k]=\sum_{n=0}^{N_a-1} f[n]e^{-j2\pi nk/N_a} \tag{2.15}$$

In the expression above, each sum will be zero for all values of *k* except *k* = 1. Further, for *k* = 1,

$$F[1]=c\left(\sin\left(\Phi-\frac{\pi}{N_a}\right)+j\cos\left(\Phi-\frac{\pi}{N_a}\right)\right) \tag{2.16}$$

Thus the estimated AOA is,

$$\hat{\Phi}=\angle F[1]-\frac{\pi}{2}+\frac{\pi}{N_a} \tag{2.17}$$

2.3 Advantages and Disadvantages of Direction Finding Systems

2.3.1 Watson-Watt

2.3.1.1 Advantages

With the appearance of low-cost, wide frequency receivers, many manufacturers realized that “stand alone” DF bearing processors could be interfaced with the new low-cost receivers at minimal cost. A well designed Watson-Watt direction finding array can be interfaced with almost any receiver with good results [8]. The Adcock antenna array’s diameter is small in size. Therefore, the array is beneficial in mobile and transportable DF applications.

Since the DF antenna tone modulation technique is AM, FM listen-through capability is excellent due to the high AM rejection of most receiver FM limiter/discriminators. Listen-through capability is also good for AM signals as a result of the fact that the DF antenna modulation tone frequencies are well below the low end of the voice spectrum and can thus be easily attenuated in the audio output channel [8].

The Adcock/Watson-Watt system is suited mainly for mobile applications especially if budgetary constraints dictate the use of low-cost receivers.

2.3.1.2 Disadvantages

The Adcock antenna array is inherently a narrow aperture. Since it is a narrow aperture, the DF resolution is affected. If a center antenna is not used, then the algorithm then suffers from an 180° phase ambiguity. A narrow aperture antenna is quite susceptible to multipath and reflection errors.

The Adcock array requires balanced sum and difference hybrids, balanced modulators, phase-matched cables, and circuits for phase or gain imbalance. All of these components can escalate the cost of the array [7].

The Adcock/Watson-Watt algorithm has a limitation on the maximum frequency. Due to the more complex electronics circuitry required by the Adcock antenna, it is not feasible to manufacture a wideband DF antenna capable of good and consistent performance at frequencies over 1000 MHz [8]. The Adcock/Watson-Watt algorithm also does not provide elevation measurements, which have greater influence over the azimuth at higher frequencies.

2.3.2 Pseudo-Doppler

2.3.2.1 Advantages

When compared to the Adcock/Watson-Watt DF system, the pseudo-Doppler systems have advantages in the areas of site error suppression, DF antenna economy, and extended high frequency performance.

Due to the circular arrangement of the antenna elements, the array can be constructed as a wide aperture array. A wide aperture array can increase AOA resolution and reduce site errors, but the size of the array then becomes an issue as the number of antenna elements increases. Because spacing between antenna elements should be $\lambda/2$ or more, as the number of elements increases, the size of the array increases. If the size of the array becomes too large, the feasibility of mobility diminishes [8].

The electronic circuitry required to implement a pseudo-Doppler system consists of GaAs FET high frequency RF switches, the necessary driver circuitry, and phase-matched cables [7]. The simpler pseudo-Doppler DF antenna array is more easily and economically designed and manufactured when compared to the Adcock/Watson-Watt system. Please note that the economic impact of the cheaper pseudo-Doppler system only applies to narrow-aperture designs. As the aperture becomes larger and more antenna elements are added, the cost of the design will increase, but they are no more as expensive as the Adcock arrays [8].

In contrast to the Adcock/Watson-Watt system, the Pseudo-Doppler algorithm should work at frequencies up to 2000 MHz and beyond due to the simplicity of the electronics associated with a pseudo-Doppler manufacturable wideband DF antenna. This allows for a greater range of applications such as cellular applications [8].

2.3.2.2 Disadvantages

Because the wide aperture pseudo-Doppler arrays can suppress site errors, the large circular arrays can limit mobility and covertness. In addition to the large arrays, the quality of the receiver needs to be more complex than the Adcock/Watson-Watt because the pseudo-Doppler receiver requires more sensitivity than an Adcock/Watson-Watt receiver, and it also needs to control the switching circuit which chooses the correct antenna element.

Another disadvantage to pseudo-Doppler systems is that the listen-through capability is a problem. Because there is a desire to obtain an accurate AOA, a high commutative switching rate is needed. When there is a high commutative switching rate, the switching rate is placed in the audio range. FM voice audio is badly distorted due to the high switching rate because the commutative process creates FM modulation at the commutative rate. AM also suffers as a consequence of the soft-commutation switches [8].

2.4 Motivation for a New Single Channel DF Method

After reviewing the above algorithms, we decided to try and develop an algorithm that is loosely based on the pseudo-Doppler system. The goal of our algorithm and system was to provide the AOA resolution of a pseudo-Doppler system or better while maintaining a small aperture mobile circular antenna array.

Our algorithm will be implemented on a software-defined radio which will take advantage of a digital implementation. It should provide equal or better performance than the algorithms or methods described above. The DF system should allow for listen-through capability which should be equal or better than the above algorithms. And last, the algorithm should be different from existing algorithms.

The Phase-locked loop (PLL) algorithm that is proposed in Chapter 3 is loosely based on the pseudo-Doppler system in that we will use a similar switched antenna array system. The advantage of the PLL algorithm is that we are able to maintain a small aperture array while increasing the AOA resolution. Therefore, the PLL algorithm is an accurate direction finding system with mobile capabilities and listen through capabilities.

Chapter 3
Phase-Locked Loop Algorithm

The previous chapters have described the state-of-the-art for single channel direction finding. Each of the existing algorithms has its own benefits and drawbacks. We now present a novel algorithm based on the use of phase-locked loops (PLLs). The PLL algorithm is similar to the Pseudo-Doppler algorithm in that the elements in a circular array are sampled using an RF switch. However, the algorithm used to estimate the AOA from the sampled antenna data is quite different. The pseudo-Doppler approach using antenna sampling is a digital implementation of an old analog concept. Here we start from the knowledge that we have the flexibility offered by digital processing to develop a more flexible and robust approach.

3.1 PLL Algorithm Development

Consider a switched circular array similar to the Pseudo-Doppler technique described in Chapter 2. However, after sampling, the signal samples are partitioned into one of N_a logical data streams. Each data stream is then processed with its own independent digital PLL implemented in software. The PLLs track the phase of the signal on each antenna which includes a phase offset common to all antennas plus the offset that is specific to each antenna θ_i. Note that the phase estimate also includes an ambiguity related to the data modulation. This ambiguity can be removed through further processing as we will discuss shortly. The switched circular array architecture is illustrated in Figure 3.1.

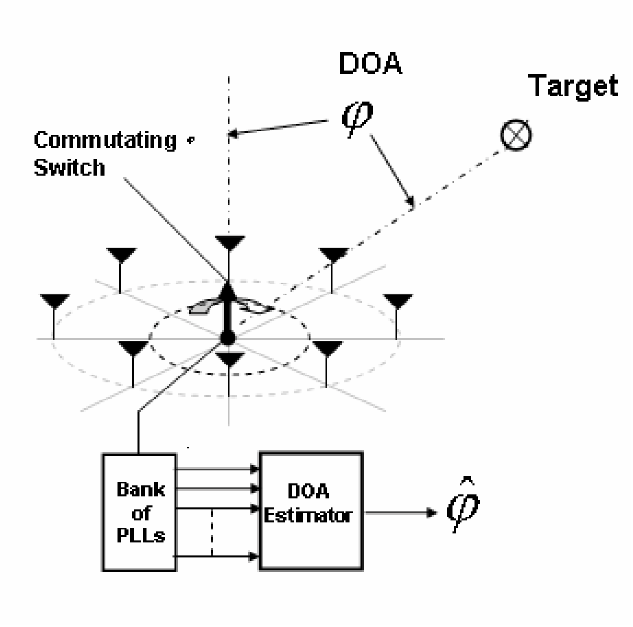


Figure 3.1 System Schematic for PLL-based AOA Estimation

The received signal going into the *i*th PLL (assuming linear modulation) is given by

$$r_i(t) = m(t)\cos\left(\omega_c t + \theta_0 + \frac{2\pi R}{\lambda}\cos\left(\frac{2\pi i}{N_a} - \Phi\right)\right) \tag{3.1}$$

The data *m*(*t*) is detected by mapping (via minimum-distance) the phase to either −*π* or *π* (in the case of BPSK) and the PLL outputs a modulo 2*π* value of the phase that lies between −*π* and *π*. Not including the data, the phase of the signal input to the *i*th PLL is given by

$$\theta_i = \theta_0 + \frac{2\pi R}{\lambda}\cos\left(\frac{2\pi i}{N_a} - \Phi\right), \qquad i = 0,1,\cdots,N_a - 1 \tag{3.2}$$

Given θ_i and provided that the phase offset θ_0 were known and there were no data modulation, we could compute the AOA, Φ, using

$$\Phi = \arccos\left(\frac{\theta_i - \theta_0}{A}\right) - \frac{2\pi i}{N_a}, \qquad \forall\ i = 0,1,\cdots,N_a - 1 \tag{3.3}$$

where *A*= 2*πr/λ* . However, due to the data modulation, the phase detected by the PLL is given by

$$\hat{\theta}_i = \theta_0 + \frac{2\pi R}{\lambda}\cos\left(\frac{2\pi i}{N_a} - \Phi\right) + n_i\pi, \qquad i = 0,1,\cdots,N_a - 1 \tag{3.4}$$

where $n_i \in \{0,\pm1,\pm2\}$, as seen in Figure 3.2, is such that $\hat{\theta}_i$ lies in $[-\pi, \pi]$ and is the closest multiple of π to the origin.

Now since $A = 2\pi r/\lambda = \pi$ (based on the design parameters), and the phase offset θ_0 lies between $[-\pi, \pi]$, in general $\hat{\theta}_i \neq \theta_i$. Therefore calculating Φ based on $\hat{\theta}_i$ would lead to incorrect AOA estimation due to the phase offset. In order to retrieve θ_i from $\hat{\theta}_i$, the sequence n_i, where $i = 0, 1, \cdots, N_a - 1$, needs to be found. If the sequence n_i, were known, and θ_o is known or can be eliminated, we can obtain an estimate of θ_i.

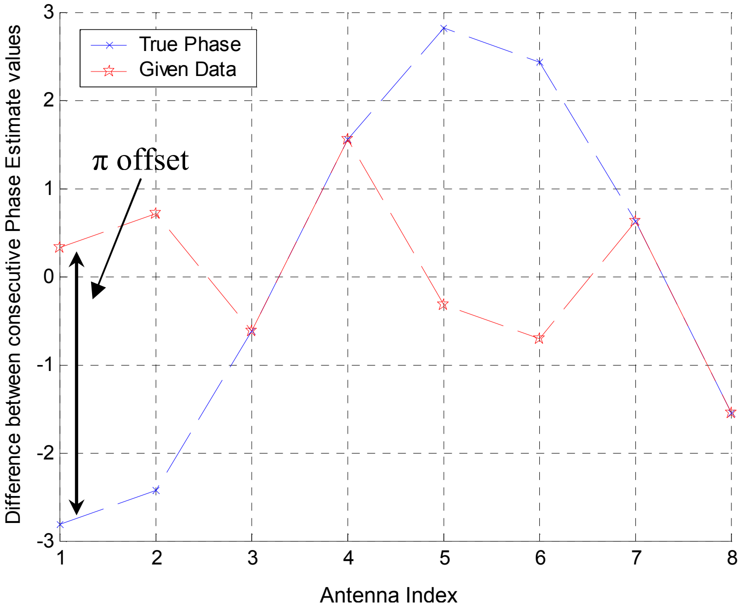


Figure 3.2 Phase values from the PLLs

A simple algorithm to solve the aforementioned problem, when $n \in \{0,\pm1,\pm2\}$, would be to store all possible (5^8) sequences $\{n_i\}$ in a table. We then add these sequences to the received $\hat{\theta}_i$, eliminate the DC offset (*i.e.*, θ_o), and exhaustively search for the most likely sequence by minimizing the squared error between the resulting vector and a set of curves with possible values for Φ. However, this is a very computationally demanding algorithm, and based on the desire for reasonable complexity and memory, this search process is not tenable. Also, if Φ

changes quickly, the receiver would need to compute this search process continuously. Hence, we desire a less computationally expensive algorithm with a smaller search-space.

In order to provide a computationally less expensive algorithm with a smaller search-space, we need to create boundaries for the algorithm. The expression corresponding to the difference between two consecutive points of $\theta_i,(i = 0,\ 1,\ \cdot\cdot\cdot,N_a - 1)$ is given by

$$\Delta\theta_i = \theta_i - \theta_{i-1} \tag{3.5}$$

$$\Delta\theta_i = \left(\theta_0 + \frac{2\pi R}{\lambda}\cos\left(\frac{2\pi i}{N_a} + \Phi\right)\right) - \left(\theta_0 + \frac{2\pi r}{\lambda}\cos\left(\frac{2\pi(i-1)}{N_a} + \Phi\right)\right) \tag{3.6}$$

$$\Delta\theta_i = \left(\frac{2\pi R}{\lambda}\cos\left(\frac{2\pi i}{N_a} + \Phi\right)\right) - \left(\frac{2\pi r}{\lambda}\cos\left(\frac{2\pi i}{N_a} - \frac{2\pi}{N_a} + \Phi\right)\right) \tag{3.7}$$

Now because $\cos(x) - \cos(y) = -2A\sin\left(\frac{x-y}{2}\right)\sin\left(\frac{x+y}{2}\right)$, we have

$$\Delta\theta_i = -2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{2\pi i}{N_a} - \frac{\pi}{N_a} + \Phi\right) \tag{3.8}$$

Let us call this the "target difference curve". From (3.8), given the value of A ($A = \pi$ and $N_a = 8$, $R/\lambda = 0.5$), we can find the maximum difference D_{max} between any two points of the actual phase θ_i. In particular, for $N_a = 8$, $D_{max} \approx 2.9$. Note that the magnitude of any point on the target difference curve cannot exceed D_{max}.

The difference between two consecutive phases *output by the PLLs* is given by

$$\Delta\hat{\theta}_i = \hat{\theta}_i - \hat{\theta}_{i-1} = -2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{2\pi i}{N_a} - \frac{\pi}{N_a} + \Phi\right) + (n_i - n_{i-1})\pi \tag{3.9}$$

We call this the "given difference curve" since it is what we obtain by taking the first difference of the output of the PLLs. Our aim is be able to find the target difference curve from the given difference curve and solve for Φ.

We see from the above equations that because we have taken the difference that the phase offset θ_0 has been eliminated, but now the difference $\Delta n_i = (n_i - n_{i-1})$ has a much larger (integer) range than n_i. However, since we know that the maximum value of $\Delta\theta_i$ is D_{max}, we can successfully eliminate several of these possibilities. Since our known values are that $D_{max} < \pi$ for $N_a = 8$ and the given value of $A = \pi$, we can eliminate all possible integer values of Δn_i, except two: one point belonging to the given difference curve (i.e., $\Delta n_i = 0$) and the other corresponding to *either* adding π or subtracting π (i.e., Δn_i = +/-1). This means that for every *i*, there exist only two possible points which could lie on the target difference curve and reduces the search space to 2^8=256.

Occasionally, the phases coming from the PLLs can differ from the target difference curve by 2π. The PLLs will only track the phase between +/- π. If there is a phase offset present, the actual phase that is being tracked by the PLLs may be greater than π or less than –π (as shown by equation (3.2)). When the true phase falls outside of [-π, +π], the PLL will map the phase back into [-π, +π] by adding or subtracting 2π. In Figure 3.3, we present an example that illustrates this point. The "theoretical" phase is presented to show what would happen if there were no phase offset present in an *ideal* system. The "expected" phase is phase data measured from each antenna element on an oscilloscope. The "actual" phase is the phase recorded on the WJ-8629A Receiver after the PLLs have converged. The "expected" and "actual phase" data has a constant phase offset from what is theoretically expected. Notice that when examining the "actual" phase data, the third antenna's phase *should* be less than –π, but the PLL forces the phase between [-π, + π] by adding 2π to the phase.

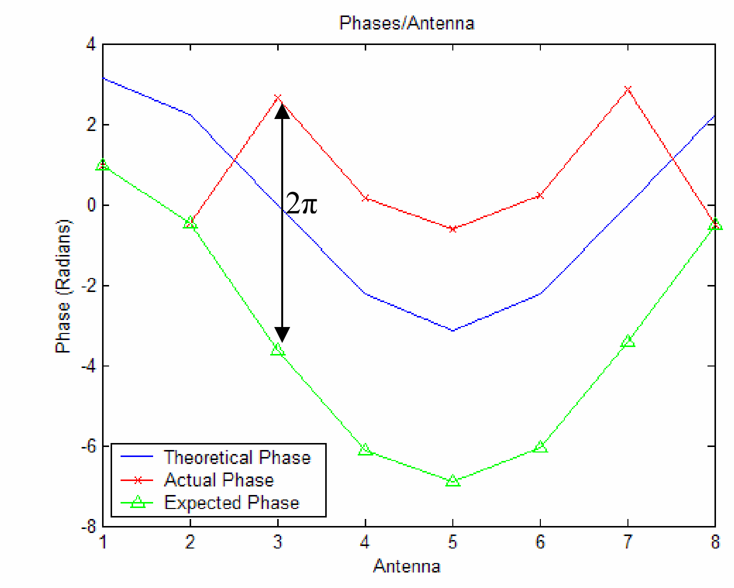


Figure 3.3 Theoretical and Experimental Phase Values at the Antenna Array

If we look at Figure 3.4 we can see the first difference curve. If the first difference of a sinusoid is taken, the result is also a sinusoid as seen in Equation (3.8). Because of the [-π, + π] boundaries that are inherent in the PLLs, the incoming signal does not match a sinusoid as seen in Figure 3.3. Therefore, the first difference of the “actual” phase data will also not be sinusoidal as seen in Figure 3.4.

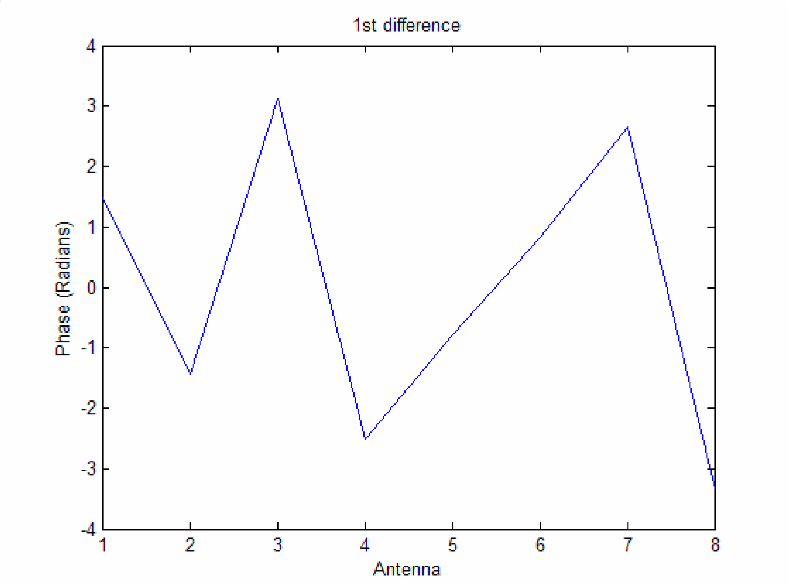


Figure 3.4 First Difference Curve

Because we base our AOA calculation from the first difference curve, if the first difference curve is not sinusoidal, then the AOA estimate will be inaccurate.

In order to correct this situation, we need to take the second difference of the “target difference curve.”

$$\Delta'\theta_i = \Delta\theta_i - \Delta\theta_{i-1} \tag{3.10}$$

$$\Delta'\theta_i = -2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{2\pi i}{N_a} - \frac{\pi}{N_a} + \Phi\right) - \left(-2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{2\pi(i-1)}{N_a} - \frac{\pi}{N_a} + \Phi\right)\right) \text{where } A = \frac{2\pi R}{\lambda} \tag{3.11}$$

$$\Delta'\theta_i = -\frac{4\pi R}{\lambda}\sin\left(\frac{\pi}{N_a}\right)\left[\sin\left(\frac{2\pi i}{N_a} - \frac{\pi}{N_a} + \Phi\right) - \left(\sin\left(\frac{2\pi(i-1)}{N_a} - \frac{\pi}{N_a} + \Phi\right)\right)\right] \tag{3.12}$$

$$\Delta'\theta_i = -\frac{4\pi R}{\lambda}\sin\left(\frac{\pi}{N_a}\right)\left[\sin\left(\frac{2\pi i}{N_a} - \frac{\pi}{N_a} + \Phi\right) - \left(\sin\left(\frac{2\pi i}{N_a} - \frac{2\pi}{N_a} - \frac{\pi}{N_a} + \Phi\right)\right)\right] \tag{3.13}$$

$$\Delta'\theta_i = -\frac{4\pi R}{\lambda}\sin\left(\frac{\pi}{N_a}\right)\left[\sin\left(\frac{2\pi i}{N_a} - \frac{\pi}{N_a} + \Phi\right) - \left(\sin\left(\frac{2\pi i}{N_a} - \frac{3\pi}{N_a} + \Phi\right)\right)\right] \tag{3.14}$$

Now because $\sin(x) - \sin(y) = 2A\sin\left(\frac{x-y}{2}\right)\cos\left(\frac{x+y}{2}\right)$, we have

$$\Delta'\theta_i = -\frac{4\pi R}{\lambda}\sin\left(\frac{\pi}{N_a}\right)\left[2\sin\left(\frac{\pi}{N_a}\right)\cos\left(\frac{2\pi i}{N_a}-\frac{2\pi}{N_a}+\Phi\right)\right] \tag{3.15}$$

$$\Delta'\theta_i = -\frac{8\pi R}{\lambda}\sin^2\left(\frac{\pi}{N_a}\right)\cos\left(\frac{2\pi i}{N_a}-\frac{2\pi}{N_a}+\Phi\right) \tag{3.16}$$

If we take the second difference of the target difference curve, we can identify when the PLLs have made a 2π correction before the first difference was taken. If the values of *consecutive* second differences exceed π *or* -π, then we have to either subtract or add π from the first difference value to correct for the forced boundaries of the PLLs.

The second difference between two consecutive first phase differences is given by

$$\Delta'\hat{\theta}_i = -\frac{8\pi R}{\lambda}\sin^2\left(\frac{\pi}{N_a}\right)\cos\left(\frac{2\pi i}{N_a}-\frac{\pi}{N_a}+\Phi\right)+\left(\Delta n_i-\Delta n_{i-1}\right)\pi \tag{3.17}$$

we call this the “given second difference curve”.

The reason that we look for consecutive second differences that exceed π or –π, is due to the first difference curve in Figure 3.4. Due to the PLL correction from antenna 3 to antenna 8 in Figure 3.3, we have spikes in the first difference at antenna 3 and antenna 8 which correspond to the PLL correction. If we take the second difference we should see consecutive peaks that exceed +/- π due to the rise and fall of those first difference peaks.

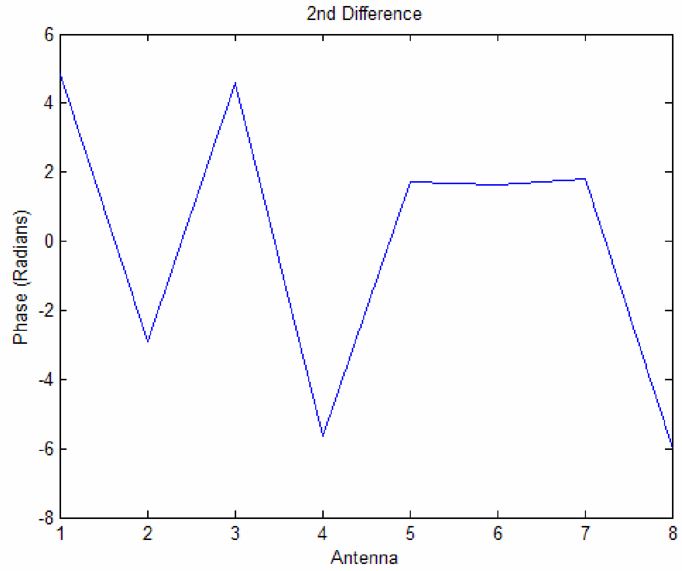


Figure 3.5 Given Second Difference Curve

If the values of consecutive second differences exceed π *or* -π, then we have to either subtract or add π to the first difference value to correct for the forced boundaries of the PLLs. In Figure 3.6, we can see how the second difference corrects the boundaries imposed due to the PLLs. Notice that the corrected first difference curve now matches a sinusoid as expected.

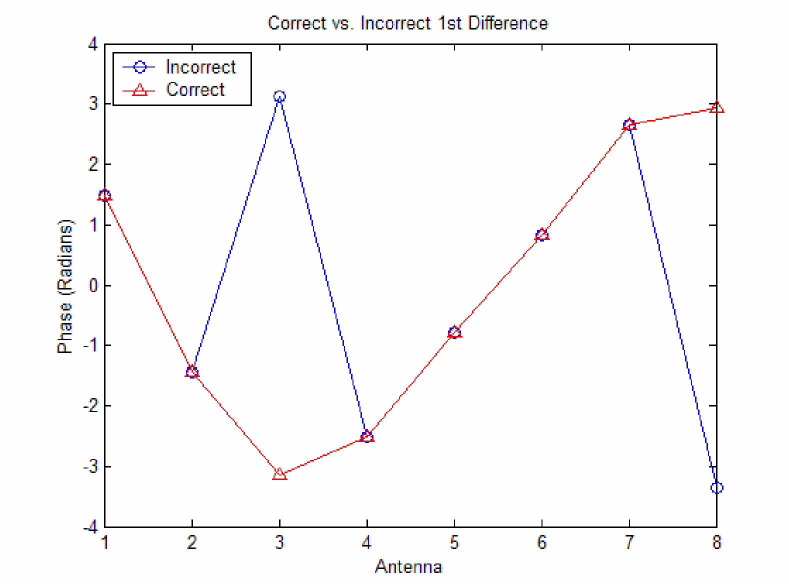


Figure 3.6 Corrected First Difference From Performing Second Difference

Now, after the first difference curve has been corrected, we can focus on calculating the estimate of the AOA. We know that the maximum value of $\Delta\theta_i$ is D_{max}, and we can successfully eliminate several of these possibilities for Δn_i as seen in Figure 3.7. Since $D_{max} < \pi$ for $N_a = 8$ and the given value of $A= \pi$, we can eliminate all integer values of Δn_i, except two: one point belonging to the given difference curve and the other corresponding to *either* adding π or subtracting π. For $N_a = 8$, since we require points corresponding to $i = (0, 1, \cdot\cdot\cdot, 7)$, we have only $2^8 = 256$ possible curves corresponding to all possible combinations of Δn_i. This search-space is much smaller and more computationally efficient than the previous search-space (5^8).

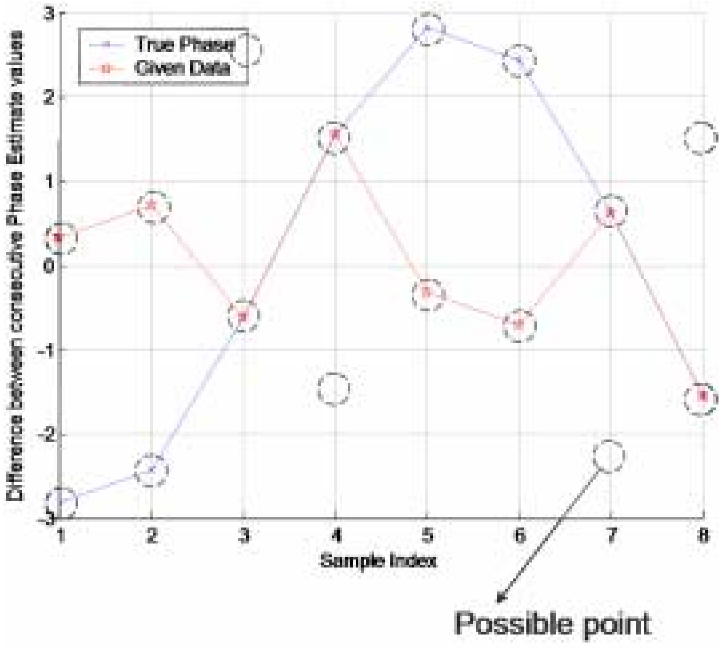


Figure 3.7 Possible Curve Fitting Points.

Since we know that the set of Δn_i that separates the given difference curve for the target difference curve can be one of 256 possible curves (Δn_i), we create a table of all 256 possible curves. Next, we divide the possible AOA, Φ, range in radians $[-\pi, \pi]$ into a number of bins, N_I, and create a target difference curve for each possible AOA. We then select one of the N_I * 256 possible target difference curves, compare them to the target difference curves, and choose the one for which the squared error is minimized. Once we have found the given difference curve, or sequence $\Delta\, n_i$, for which the squared error is minimized, we can use the target difference curve to compute the angle of arrival, Φ.

The previous procedure allows us to find the sequence $\{\Delta n_i\}$ that results in a target curve that is closest in the square error sense to the original data. Although, we posit several possible values of **Φ** to do this, we will solve directly for **Φ** after $\{\Delta n_i\}$ have been determined. After removing the $\{\Delta n_i\}$ terms, the target difference curve can be represented as the following vector:

$$f[n] = \begin{bmatrix} -2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{2\pi}{N_a}-\frac{\pi}{N_a}+\Phi\right) \\ -2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{4\pi}{N_a}-\frac{\pi}{N_a}+\Phi\right) \\ \cdots \\ -2A\sin\left(\frac{\pi}{N_a}\right)\sin\left(\frac{2\pi i}{N_a}-\frac{\pi}{N_a}+\Phi\right) \end{bmatrix} \tag{3.18}$$

The FFT of this vector is

$$F[k] = \sum_{n=0}^{N_a-1} f[n]e^{-j2\pi nk/N_a} \tag{3.19}$$

In the expression above, in the absence of noise, *F[k]* will be zero for all values of *k* except $k = 1$. Further, for $k = 1$,

$$F[1] = c\left(\sin\left(\Phi-\frac{\pi}{N_a}\right)+j\cos\left(\Phi-\frac{\pi}{N_a}\right)\right) \tag{3.20}$$

Thus the estimated AOA is found as,

$$\hat{\Phi} = \angle F[1]-\frac{\pi}{2}+\frac{\pi}{N_a} \tag{3.21}$$

3.2 Simulation Results

Now that we have defined our phase-locked loop algorithm for single channel direction finding, we can examine the performance of our algorithm. In Figure 3.8, we plot the AOA accuracy of the PLL algorithm in an AWGN channel in terms of average estimated AOA and average AOA error. The algorithm was simulated at an SNR of 10 dB, a PLL update parameter of $\mu = 0.1$, and 100 simulation runs per angle-of-arrival.

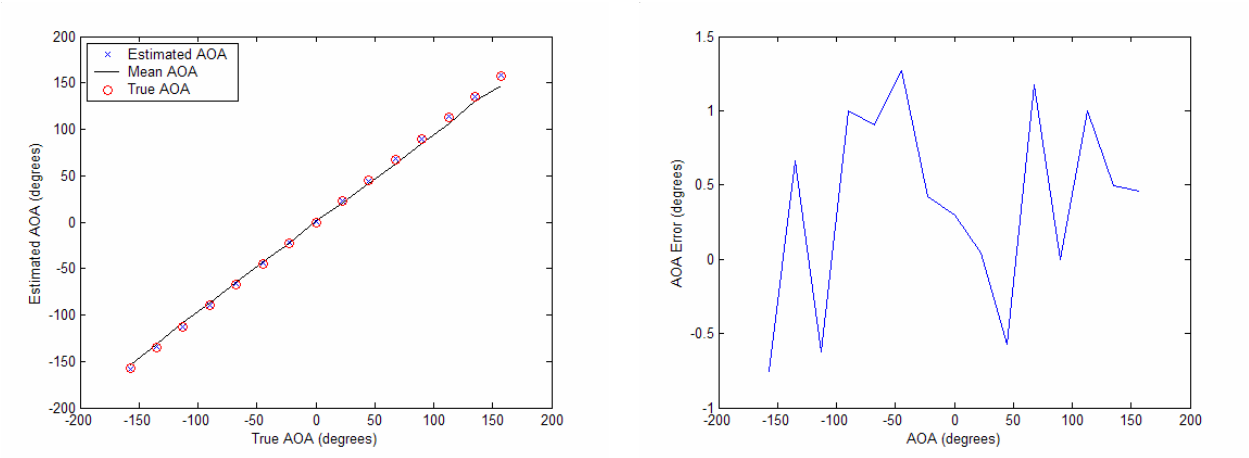


Figure 3.8 AOA Accuracy and AOA Error in an AWGN channel

In Figure 3.8, the mean estimated AOA is plotted versus the true AOA. In the left plot, the circles show the True AOA under test. The black line is the mean estimated AOA that is being tracked by the algorithm. The blue ‘x’ is the last estimated AOA that was tracked by the algorithm. The mean differs from the last estimated value due to the convergence time of the PLL. Thus, by looking at the AOA error from the last sample after PLL convergence (Figure 3.8 – right), we can see that the PLL algorithm is within ~1.3 degrees of the true AOA.

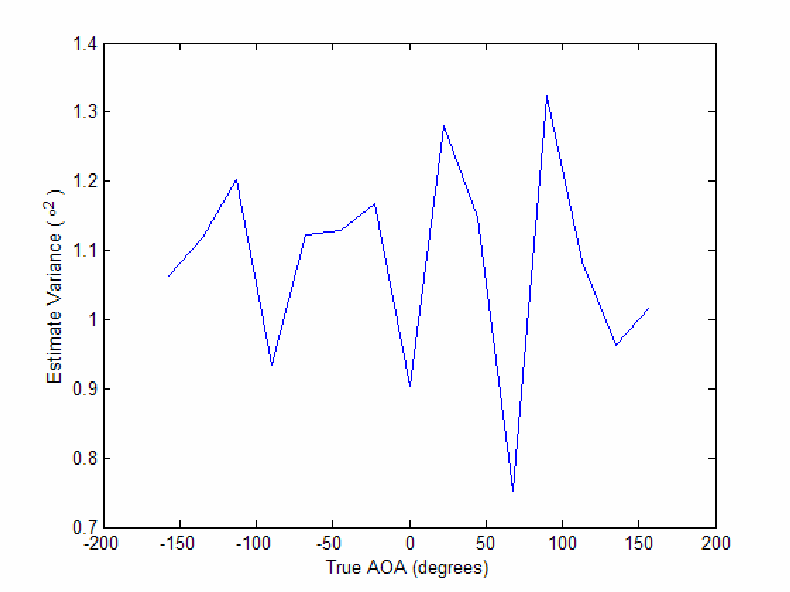


Figure 3.9 Error Variance After PLL Convergence

Figure 3.9 plots the error variance of the PLL algorithm after the PLLs have converged over 100 simulation runs at each AOA. We can see from the figure that the variance is very small ranging from 0.75 ^{o2} to 1.32 ^{o2}.

In Figure 3.10, the effects of SNR on the algorithm are shown. Specifically, we plot the mean estimated AOA after the PLLs have converged for a true AOA of 45°. We can see that even at an SNR below 2 dB, the PLL algorithm still is accurate to within 1.5 degrees.

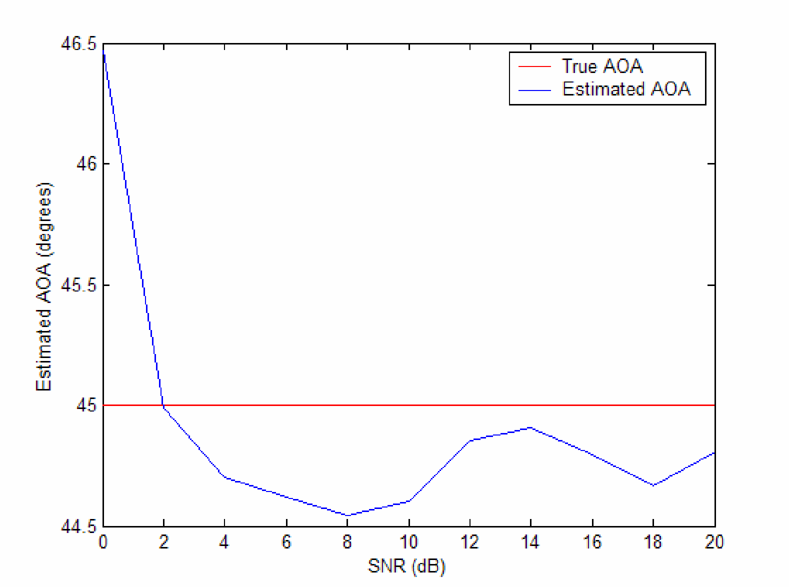


Figure 3.10 Mean AOA accuracy vs. SNR in AWGN channel

Notice that at SNR values > 2 dB, we have an AOA estimate that is within 0.5 degrees. As is common in digital implementations, when the SNR decreases below a certain threshold, the AOA accuracy will diverge quickly.

In the previous simulations, we estimated the AOA in cases where there were no phase offsets. By using equation (1), we will choose a specific AOA of 45 and -65 degrees and increase the phase offset, θ_0, from 0 to 2π. Figure 3.11 illustrates how an induced phase offset can affect the PLL algorithm accuracy when estimating the AOA.

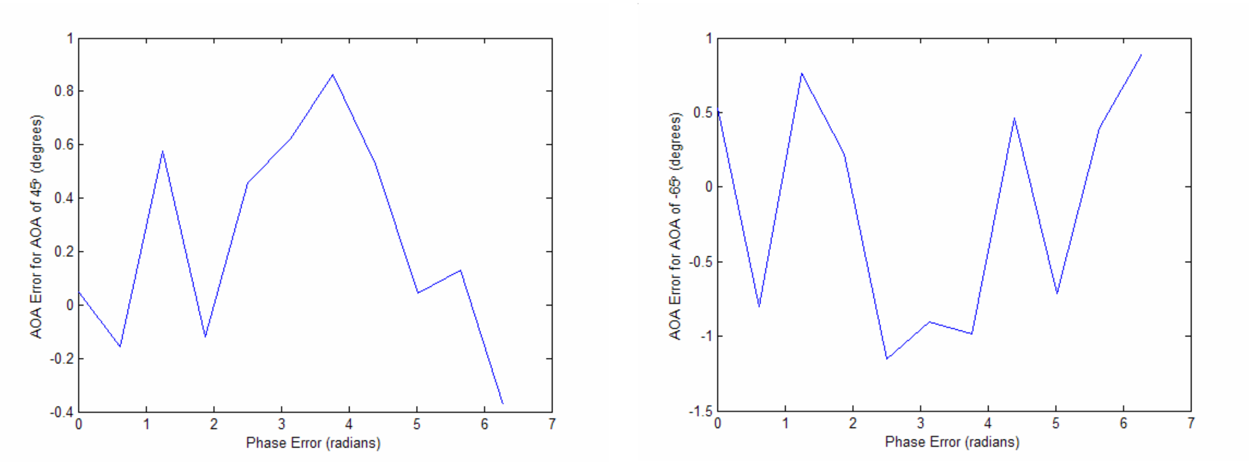
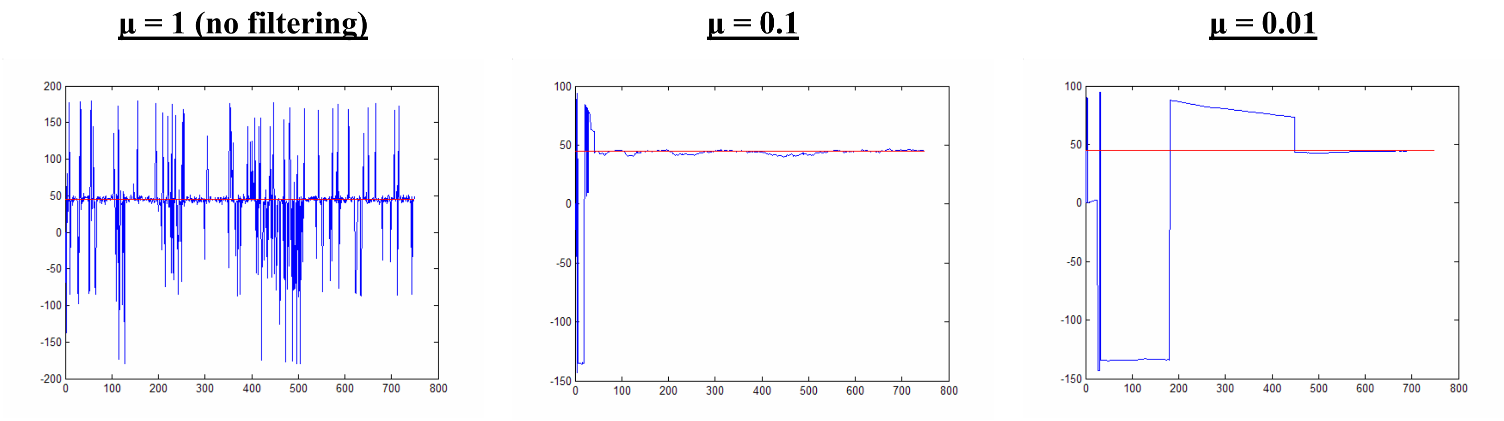


Figure 3.11 AOA Error vs. Phase Offset

At both 45 and -65 degrees in the above figure, there is minimal impact to the accuracy of the AOA as the phase error or phase offset is increased from 0 to 2π.

The previous discussion ignored the impact of PLL filtering. In the previous examples, we have assumed a PLL update parameter of $\mu = 0.1$. If the update parameter $\mu = 1$, then there is no filtering applied to the PLLs. As μ gets smaller, it will take longer for the PLLs to converge, but the AOA will achieve higher accuracies due to less noise in the PLL outputs. Specifically, since the AOA and phase estimates do not change instantaneously, we can filter the PLL outputs in time to improve performance. Figure 3.12 (far left) demonstrates the transient behavior of the AOA estimation algorithm at an SNR of 10 dB. Specifically, before the PLLs have converged, the AOA estimate varies wildly. This is not surprising since the phase estimates themselves are quite variable without filtering.

To illustrate the effect of filtering the PLLs in order to estimate an AOA of 45 degrees, Figure 3.12 examines three different scenarios: $\mu = 1$ (far left), $\mu = 0.1$ (middle), and $\mu = 0.01$ (far right). From Figure 3.12, due to the unfiltered PLLs, the estimated AOA fluctuates wildly. As we apply filtering to the PLLs, we can see that the estimated AOA will fluctuate initially, but when the PLLs lock onto the phase as shown in Figure 3.13, the estimated AOA converges to the desired AOA of 45 degrees. This is a common phenomenon with adaptive filtering; reducing the update parameter slows convergence but improves performance after convergence.


μ = 1 (no filtering)

μ = 0.1

μ = 0.01

Figure 3.12 PLL Filtering with Update Parameter of μ = 1, 0.1, and 0.01

Figure 3.13 displays an example of the PLL tracking on the first antenna with and without filtering. When no filtering is applied, the PLL will not converge to the correct phase as seen. Because the PLL does not track to the correct phase, the estimated AOA will fluctuate around the True AOA as was shown in Figure 3.12.

When filtering is applied with μ = 0.1, we can see that the PLL will track quickly to a phase value, but there are still significant variations even though the phase has “locked”. Because the PLLs track quickly, the estimated AOA is found quickly, but there may be small errors in the estimated AOA as seen in Figure 3.12.

When filtering is applied with μ = 0.01, we can see that the PLL will track more slowly than when μ = 0.1, but the PLL can track the phase more precisely. There are not any significant variations, but we pay a price in reduced tracking speed. Because the PLLs are tracking slowly but accurately, the estimated AOA will take longer to converge but there will be an improvement in AOA accuracy as seen in Figure 3.12.

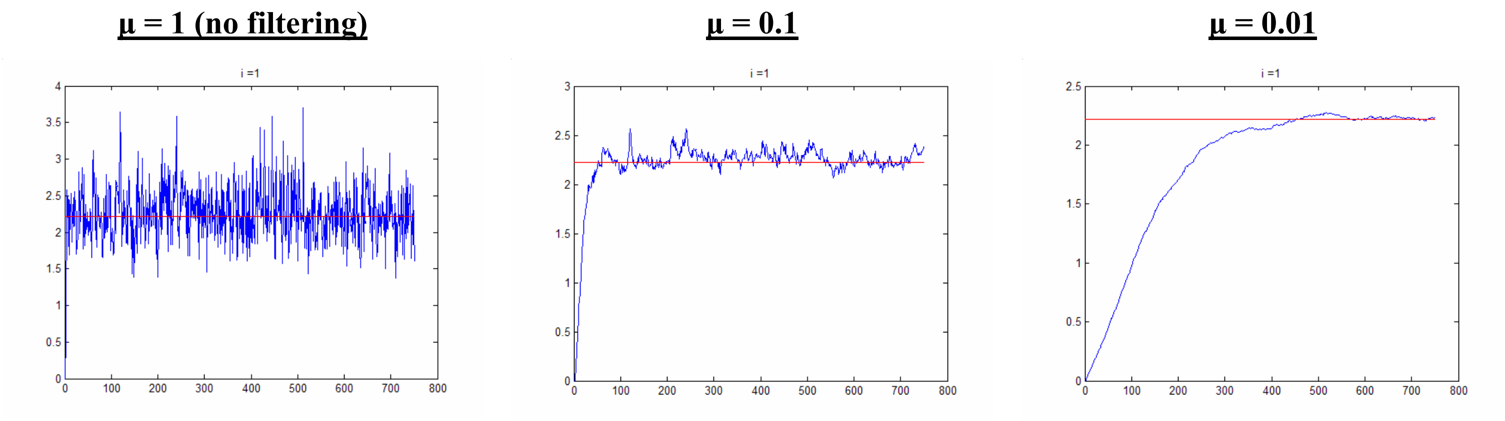


Figure 3.13 First PLL Convergence for different values of μ.

In Figure 3.14, we can see that by filtering the estimated AOA (in addition to filtering the PLL outputs) we can get smooth out large jumps in the AOA that occur at lower SNRs. The only drawback to AOA filtering is that it can take longer to converge to the correct AOA. Figure 3.14 examines the effect of AOA filtering on a signal with an AOA of 45 degrees.

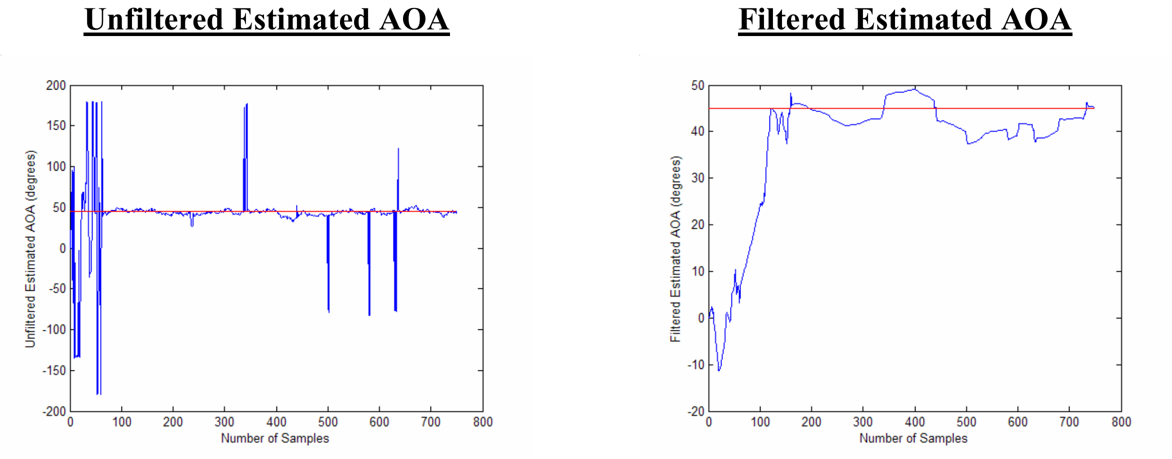


Figure 3.14 Filtering performed on AOA at SNR of 0.75dB.

We can achieve an AOA accuracy of 5 degrees after 150 samples at an SNR of 0.75dB and μ = 0.1. We can see that the filtering of the AOA smoothes out the large jumps in the estimated AOA unlike the unfiltered case shown in Figure 3.14 (left).

All of the above simulations deal with the accuracy of the estimated AOA in an AWGN channel. Most direction finding systems require tracking an emitter not just in an AWGN channel, but also in a Rayleigh fading channel.

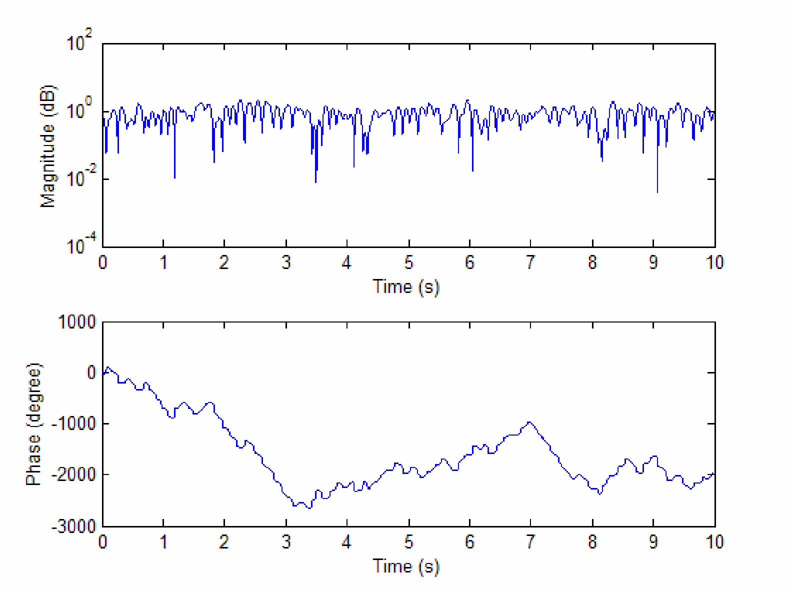


Figure 3.15 Rayleigh Fading Channel – Magnitude and Phase

In Figure 3.15, we have created a Rayleigh Fading channel with a Doppler frequency of 5 Hz to test our PLL algorithm at an SNR of 20 dB. Figure 3.16 shows that the PLL algorithm does produce an accurate estimated AOA in a fading channel represented in Figure 3.15 with an SNR of 20 dB and μ = 0.01.

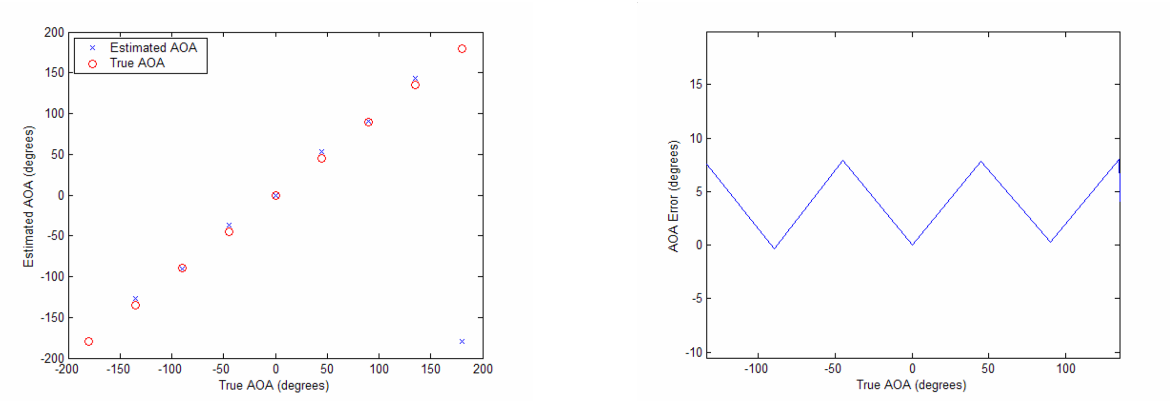


Figure 3.16 Estimated AOA vs. True AOA in a Rayleigh Fading Channel

We can see that at the True AOA of -180 degrees and 180 degrees that the PLL algorithm has estimated the AOA 360 degrees away, which is still correct. When examining the AOA error, we can see that there is a maximum error of 8 degrees due to the fading channel.

In Figure 3.17, the two-ray multipath model is illustrated [10]. The two-ray model has a single emitter, and the array sees two paths from that emitter. One of the paths is the direct line-of-sight (LOS) and the other is a multipath component. The multipath component can also be viewed as a second emitter that results in co-site interference. No fading is considered in this model.

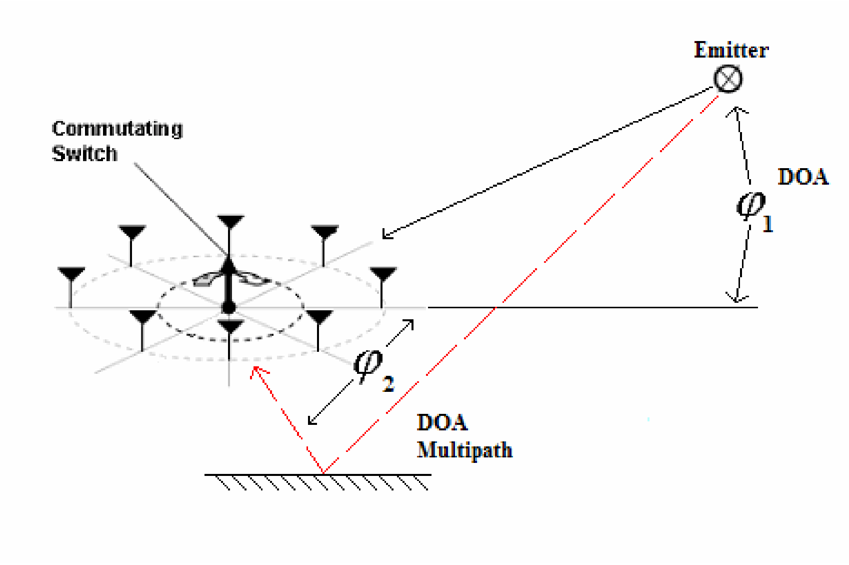


Figure 3.17 Two-Ray Multipath Model

As the power ratio of AOA #2 vs. AOA #1 increases linearly, the estimated AOA does not deviate from the emitter (AOA #1) until the multipath component power (AOA #2) approaches the emitter power as seen in Figure 3.18. When the multipath component power is more dominant than the emitter, the PLL algorithm will track the multipath component instead of the emitter. The reason the PLL algorithm tracks the more dominant AOA is due to the theory of phase-locked loops. Once the PLLs lock onto a phase, the tracked AOA will not deviate unless the PLLs track onto another phase due to the power of another emitter, or if the PLLs loose track due to the amount of noise present.

The power of a multipath component should not exceed the power of a LOS signal due to laws of propagation. However, if both signals are NLOS signals with one coming from the LOS direction, the second signal may exceed the power of the first.

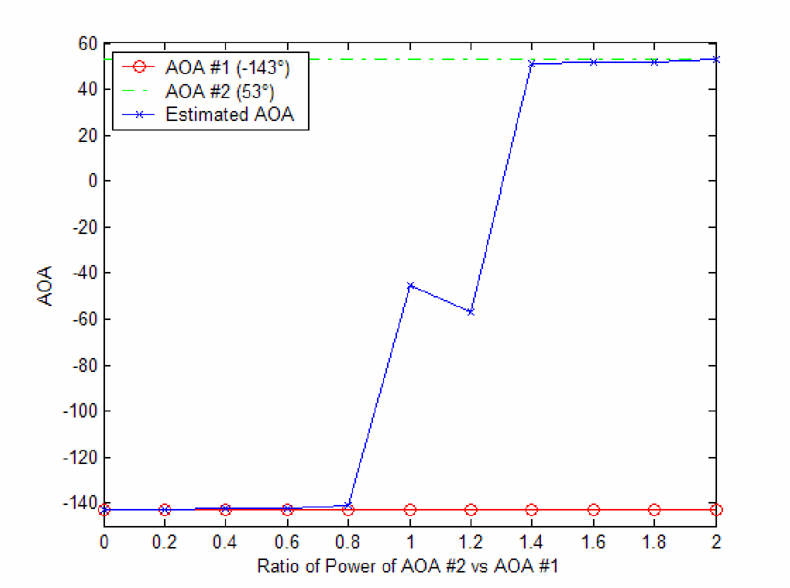


Figure 3.18 Estimated AOA for Two Dominant Multipath Components.

From Figure 3.18, we can see that the PLL algorithm can track the dominant AOA accurately until the multipath or second emitter has a power level equal to or greater than the first emitter. As the power ration hovers around the value of 1, we roughly have equal power between the emitter and the multipath component. The algorithms accuracy starts to degrade when the signals have equal power. Once one of the power levels becomes dominant, the PLL algorithm will track to that AOA.

This development is very promising because the PLL algorithm can naturally eliminate multipath components if the line-of-sight (LOS) component has greater power than a single multipath component. This algorithm can provide accurate results in a multipath environment when the constructive interference of the power from multipath components does not exceed the power from the LOS component.

Now that we have thoroughly examined the PLL algorithm's performance in different situations and in different environments, we can compare the results seen above to the results of the other DF algorithms presented in Chapter 2.

Chapter 4

Performance of Alternative Single Channel DF Techniques

In Chapter 2, we discussed the two major single channel direction finding approaches, the Watson-Watt and the Pseudo-Doppler techniques. Further, we discussed their relative merits and demerits. We would now like to directly compare the performance of the algorithms with the proposed PLL approach examined in Chapter 3. In this chapter we will present simulation results that explore the other two single channel direction finding algorithms and compare them with the proposed approach.

Specifically, for each algorithm discussed in Chapter 2, we will examine the AOA accuracy of the algorithm as a function of the signal-to-noise ratio (SNR), the effect of cosite interference or multipath, and the effects of a Rayleigh fading channel. Note that all simulations include AWGN.

4.1 Performance of Adcock/Watson-Watt Technique

The first algorithm that will be examined is the Watson-Watt algorithm, which uses an Adcock antenna array with d = 0.25 ($d = R/\lambda$) as described in Chapter 2. Due to the sin function in the numerator and the cosine function in the denominator, presented in Equation 2.3 in Chapter 2 and restated below,

$$\begin{aligned}\hat{\Phi}(t) &= \arctan\left(\frac{y(t)}{x(t)}\right)\\ &= \arctan\left(\frac{2j*m(t)\sin\left(\frac{2\pi R}{\lambda}\sin(\Phi)\right)}{2j*m(t)\sin\left(\frac{2\pi R}{\lambda}\cos(\Phi)\right)}\right)\\ &= \arctan\left(\frac{\sin\left(\frac{2\pi R}{\lambda}\sin(\Phi)\right)}{\sin\left(\frac{2\pi R}{\lambda}\cos(\Phi)\right)}\right)\\ &\approx \arctan\left(\frac{\sin(\Phi)}{\cos(\Phi)}\right)\end{aligned} \tag{4.1}$$

there is a natural error or inherent bias associated with the Watson-Watt algorithm. As seen in Figure 4.1, the bias associated with the Watson-Watt algorithm is sinusoidal with regular maxima. Without AWGN being added to the system, we still have an AOA error that can be as large as 7 degrees.

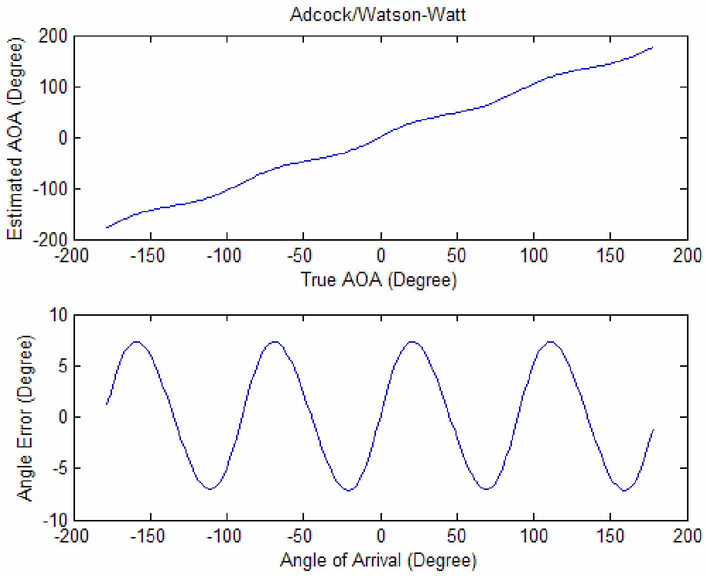
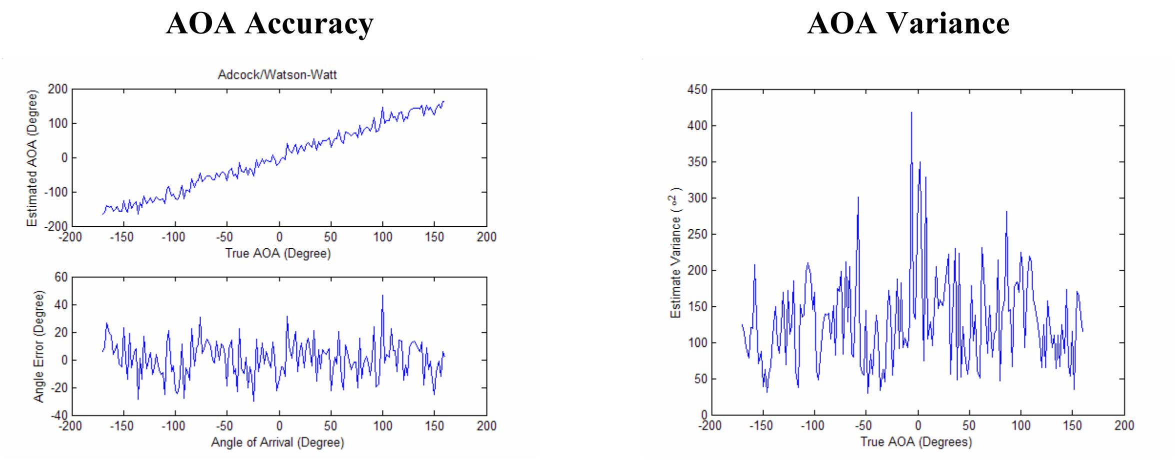


Figure 4.1 Adcock/Watson-Watt Algorithm's Inherent Bias

By looking at Figure 4.1, we can see that the AOA error presented in the lower graph is sinusoidal. The sinusoidal error curve results from the arctangent of the amplitude differences being taken to calculate the AOA as seen in Equation 4.1.


AOA Accuracy

AOA Variance

Figure 4.2 Adcock/Watson-Watt AOA Accuracy at an SNR of 10 dB

We find that even at an SNR of 10 dB the average error is still approximately 20 degrees as seen in Figure 4.2. The variance of the Watson-Watt algorithm is not promising. The variance can have a maximum > 300 degrees.

Figure 4.3 presents the mean accuracy (200 simulation runs) of the AOA estimate when the true AOA is 45 degrees and the SNR varies from 0 to 20 dB. The emitter's AOA is at a location of 45 degrees with respect to the East Antenna. The AOA of 45 degrees was chosen due to the accuracy of the Watson-Watt algorithm. Recall from Figure 4.1, that the Watson-Watt algorithm displays its best accuracy at an AOA of 45^o. Thus, the results in Figure 4.3 represent optimistic results.

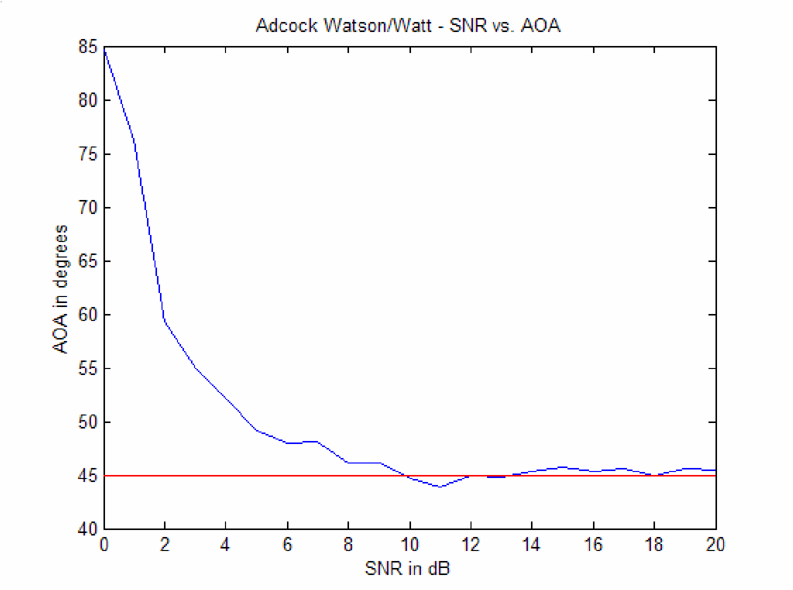


Figure 4.3 Watson-Watt AOA Accuracy vs. SNR

As the SNR increases, the accuracy of the algorithm improves, but as the SNR decreases below 10 dB, we can see that the error can increase up to 40 degrees. Recall from Figure 3.10 in Chapter 3 that our PLL algorithm deviated roughly 3.5 degrees at SNRs lower than 5 dB. From the above figure, we can see that the PLL algorithm provides a significant improvement in AOA accuracy, particularly at lower SNRs. However, it should be noted that the PLL algorithm uses eight antennas as opposed to four.

When examining the impact of resolvable multipath on the Watson-Watt algorithm, we have chosen to model a two-ray multipath scenario as presented in Figure 3.16 in Chapter 3. Specifically, the first signal is received at an AOA of -143^{o} with respect to the East Antenna and the multipath signal is received at an AOA of 53^{o} with respect to the East Antenna. The true AOA corresponds to the first path which is at -143^{o}.

In Figure 4.4, the Watson-Watt algorithm at an SNR of 10 dB and r/λ = 0.25 is shown as the power ratio of AOA #2 to AOA #1 increases. The AOA estimate is fairly accurate when the LOS component is dominant with perhaps a few degrees of error. However, when the multipath component is dominant, the AOA estimate clearly follows the multipath. When the received powers for the two paths are approximately the same, the algorithm starts to deviate from the true AOA and then switches over to the multipath AOA when it starts to dominate.

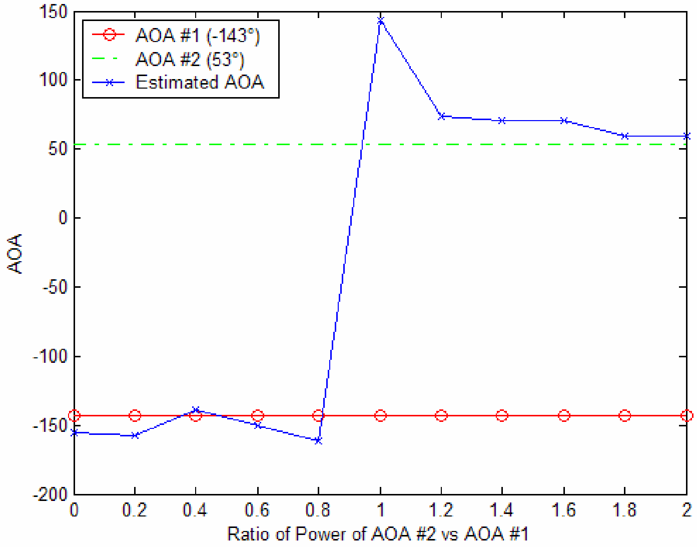


Figure 4.4 Watson-Watt Multipath Effects with Two Emitters

Figure 4.5 shows two results of the accuracy of the true AOA vs. the estimated AOA of the Watson-Watt algorithm in a Rayleigh fading channel with a Doppler frequency of 5 Hz at an SNR of 30 dB. Since the Watson-Watt algorithm is an amplitude-based direction finding algorithm, it has trouble providing an accurate estimate of the AOA in a fading channel.

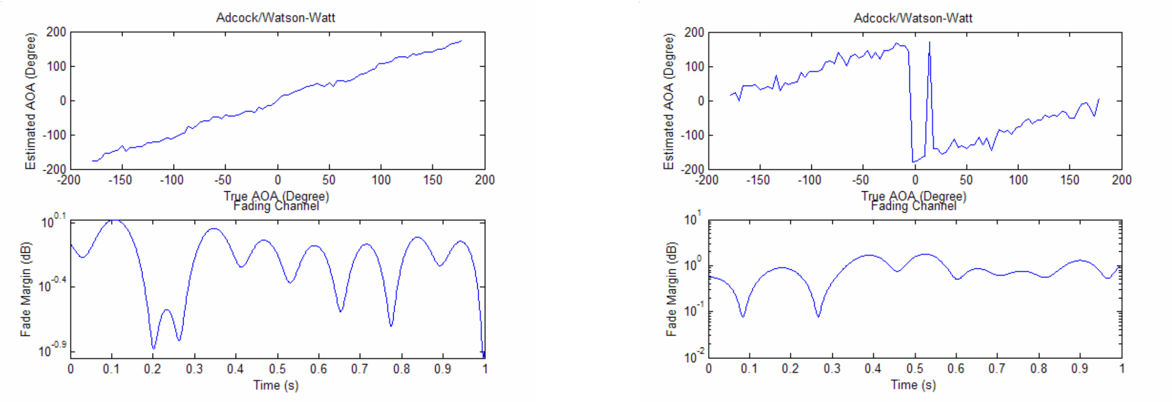


Figure 4.5 Watson-Watt Accuracy with Rayleigh Fading Channel at SNR = 30 dB.

In Figure 4.5, we can see that the Watson-Watt algorithm can produce an AOA that is 180 degrees out of phase from the true AOA in a fading channel. When examining the Watson-Watt algorithm in a fading channel (Doppler frequency of 5 Hz at an SNR of 30 dB) with respect to time in Figure 4.6, we can see that the estimated AOA varies between the true AOA of 90^{o} and an AOA that is 180 degrees out of phase. This AOA variation can be attributed to the phases induced by the fading channel. Because the changing phase affects the adding and subtracting of elements, there will be a 180 degree phase ambiguity when we take the arctangent of a negative tangent (Equation 2.3) for the example shown in Figure 4.6.

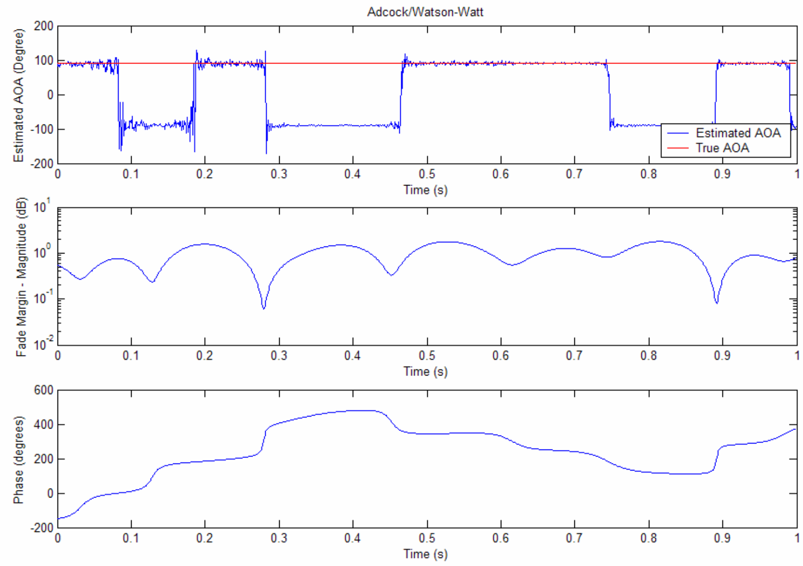
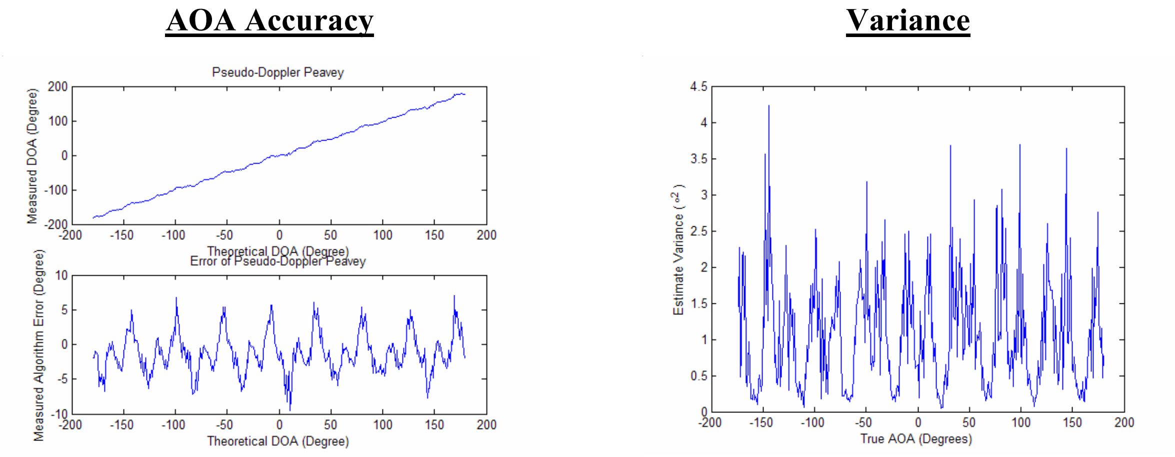


Figure 4.6 Watson-Watt Accuracy with Rayleigh Fading Channel at SNR = 30 dB.

Due to the biased nature of the Watson-Watt algorithm, its poor performance with lower SNRs, multipath, and Rayleigh fading proves that it is an inferior algorithm for our project when compared to the results in Chapter 3.

4.2 Performance of Pseudo-Doppler Algorithm

The next algorithm that was examined using MATLAB simulations was the Pseudo-Doppler algorithm proposed by David Peavey and Dr. Tokunbo Ogunfunmi. Their algorithm employs a feedback system via a vector modulator in the IF of a single receiver and an adaptive algorithm to control it [3]. In Figure 4.7 the simulated performance of the Pseudo-Doppler is plotted for an 8-element array. We can see that the algorithm provides a more accurate AOA estimate than the Watson-Watt algorithm. This is not surprising since it has twice as many antenna elements. At an SNR of 10 dB, the Pseudo-Doppler algorithm has a maximum error of about 5 degrees from the true AOA.


AOA Accuracy

Variance

Figure 4.7 Pseudo-Doppler AOA Accuracy at SNR of 10 dB

As we did with the previous algorithms, the Pseudo-Doppler algorithm was tested at several different SNRs to examine at what SNR the algorithm starts to deviate significantly from the true AOA. In Figure 4.8, the algorithm provides a mean AOA (100 simulation runs) accurate to roughly 0.8 degrees at SNRs lower than 10 dB for an AOA of 45 degrees. Once the SNR is greater than 10 dB, the algorithm estimates the AOA to within 0.4 degrees. The AOA of 45 degrees was chosen due to the accuracy of the pseudo-Doppler algorithm. As seen in Figure 4.7, the algorithm provides an accurate AOA at 45 degrees.

Because the AOA of 45 degrees was chosen due to the accuracy of the pseudo-Doppler algorithm, a second AOA was chosen to demonstrate its inaccuracies. At an AOA of 80 degrees, we can see that the pseudo-Doppler algorithm does not estimate the AOA accurately even at higher SNRs. We can see that if we look at the mean AOA, that we have a maximum deviation of ~ 8.5 degrees. In Figure 4.7, we can see a deviation of ~6 degrees at an SNR of 10 dB.

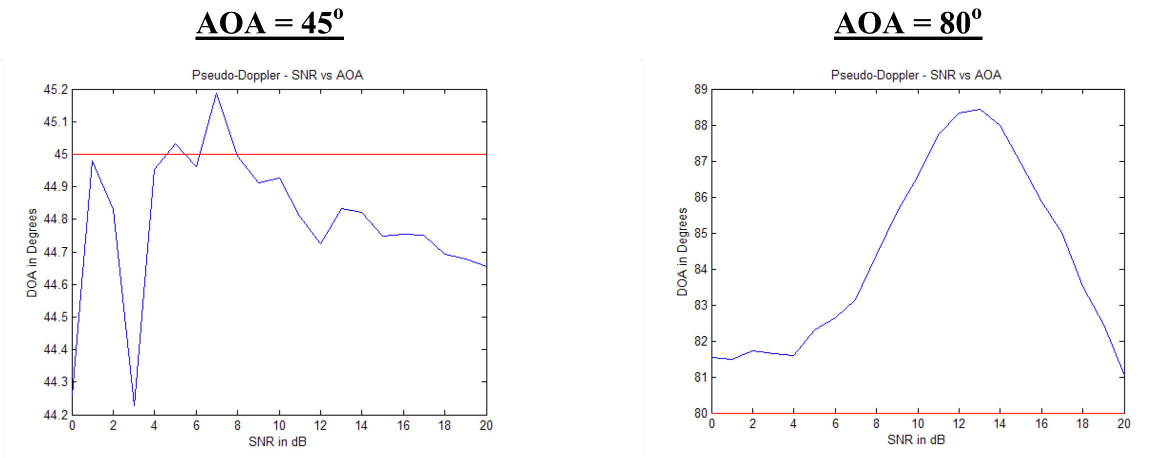


Figure 4.8 Pseudo-Doppler Algorithm Accuracy vs. SNR at Different AOA Values

The pseudo-Doppler algorithm can estimate the AOA better than the Watson-Watt algorithm at lower SNRs. Even though the pseudo-Doppler algorithm proves to be more accurate than the Watson-Watt algorithm simulated in the above sections, it is still less accurate overall at estimating an AOA than the PLL algorithm.

As we saw in Chapter 3, the PLL algorithm provides an improved AOA at lower SNRs and at higher SNRs. The PLL algorithm can estimate the AOA to within 3 degrees at SNRs lower than 5 dB. The PLL algorithm proves that it is better suited for low SNR direction finding environments.

In Figure 4.9, we can study the effects of multipath on the pseudo-Doppler algorithm. While it is an improvement over the above algorithms with respect to AOA accuracy, we can see that as the multipath component becomes more dominant around 0.6, the AOA accuracy starts to deviate from the true AOA of -143 degrees.

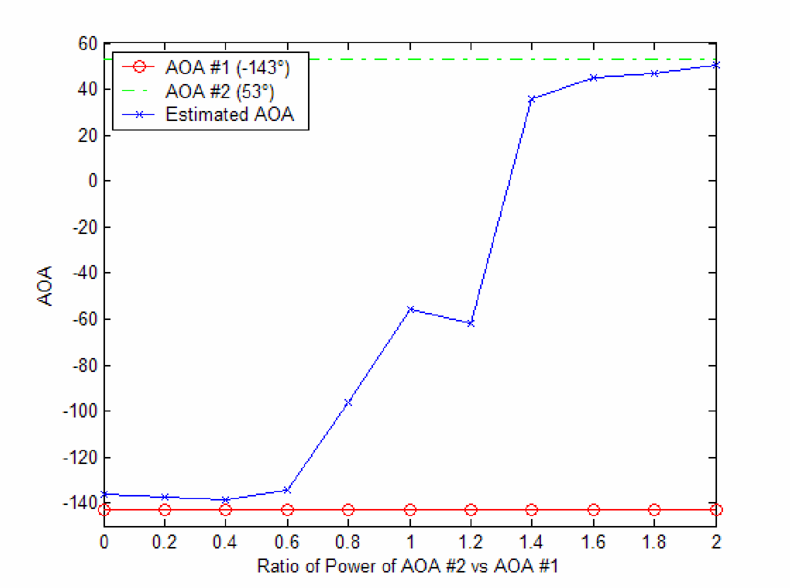


Figure 4.9 Pseudo-Doppler Multipath Effects with Two Emitters

When comparing Figure 4.9 to Figure 3.17 in Chapter 3, we can see that the pseudo-Doppler algorithm is not as accurate as the PLL algorithm when either of the emitters are dominant. Also, the pseudo-Doppler algorithm starts to deviate from the true AOA when the ratio of the power of the multipath to the power of the true AOA is 0.6. The PLL algorithm starts to deviate when the ratio is 0.8. Both algorithms provide a better response than the Watson-Watt algorithm which tends to “overshoot” the AOA when the ratio is nearly 1.

The pseudo-Doppler algorithm was then tested in a Rayleigh fading channel with a Doppler frequency of 5 Hz as tested in the above examples. The algorithm was able to track the AOA as seen in Figure 4.10. The algorithm was then tested with a Doppler frequency of 50 Hz. We can see that if the sampling at each antenna and the array switching rate are kept constant, the algorithm has trouble tracking the AOA in a fading channel with a higher Doppler frequency. However, because the pseudo-Doppler algorithm can accurately track the AOA in a fading channel for a Doppler frequency of 5 Hz, it gives itself an advantage over the Watson-Watt algorithm.

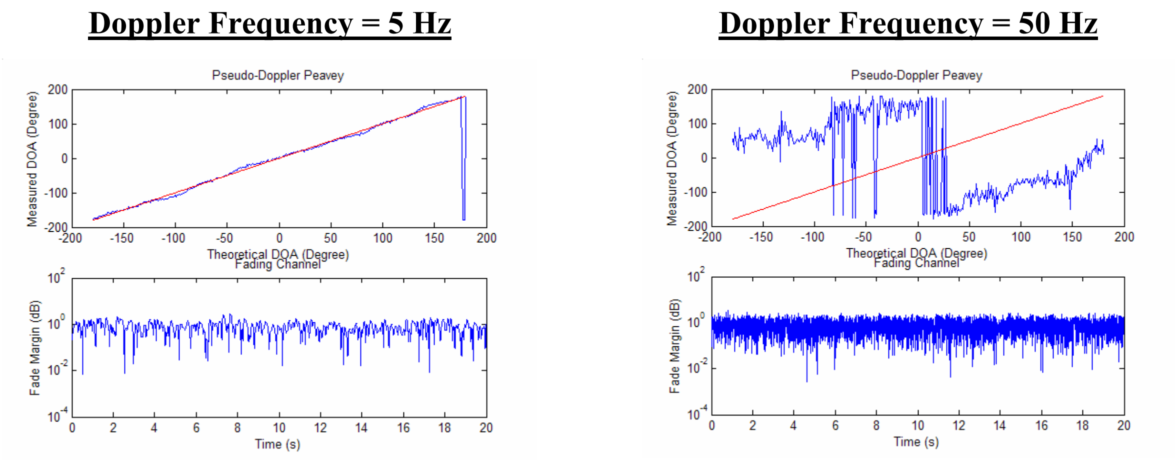


Figure 4.10 Pseudo-Doppler AOA Accuracy in Rayleigh Fading Channel

4.3 Comparison of Algorithms

By examining the above simulations, The Watson-Watt is not accurate at lower SNRs but can survive in a Rayleigh fading channel if the fades aren't too deep. The pseudo-Doppler algorithm provides an increase in accuracy for AOA estimation in AWGN, multipath, and fading channels. Using some of the strengths of the Pseudo-Doppler, such as a commutated switched array and eight antenna elements, we were able to develop our new PLL algorithm. We can see in Chapter 3 that the PLL algorithm provides an enhancement in accuracy at different SNRs but especially at lower SNRs as seen in Figure 4.11.

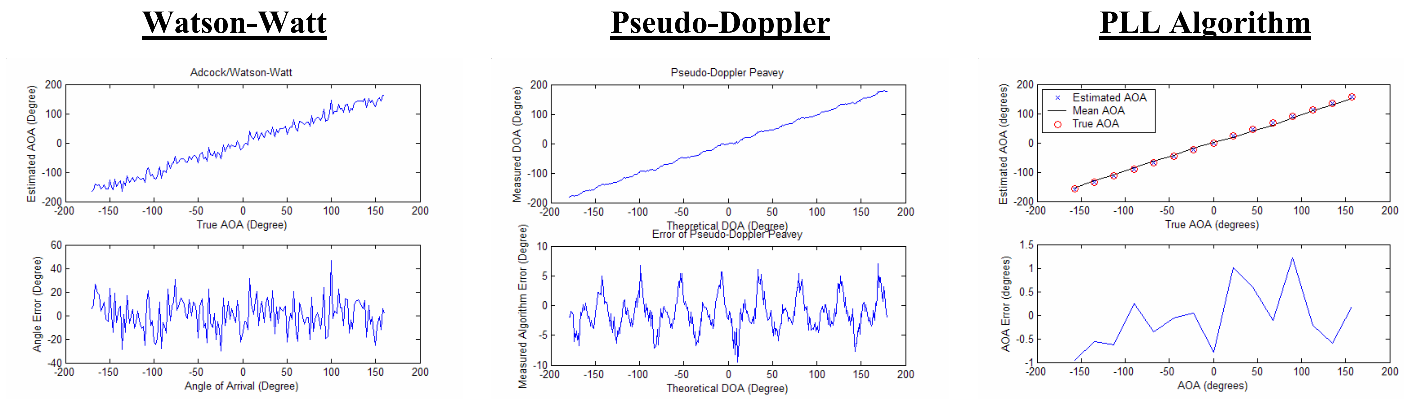


Figure 4.11 Comparison of Average AOA Accuracy and Average AOA Error for Different Algorithms (SNR = 10 dB)

In Figure 4.12, our PLL algorithm also shows an improvement when comparing the AOA variance for each algorithm. We can see that the variance of the Watson-Watt algorithm is large at an SNR of 10 dB. Both the Pseudo-Doppler and PLL algorithm have small AOA variances, but the PLL algorithm's variance is half as large as the pseudo-Doppler algorithm.

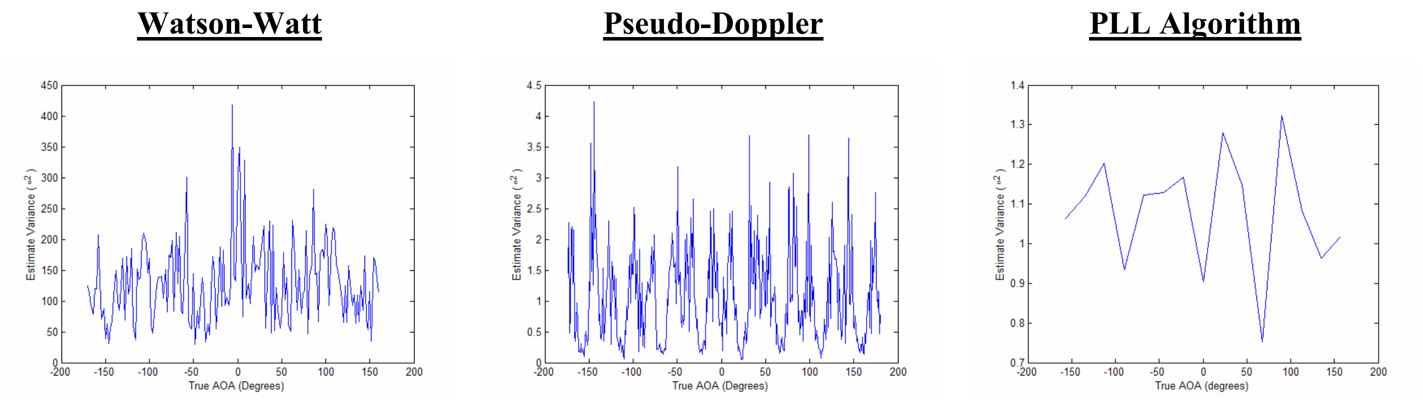


Figure 4.12 Comparisons of Variances of the Three Algorithms.

In order to grasp the impact of the SNR on the AOA estimate variance, Figure 4.13 displays plots that show the SNR required to get the variance below 1 degree.

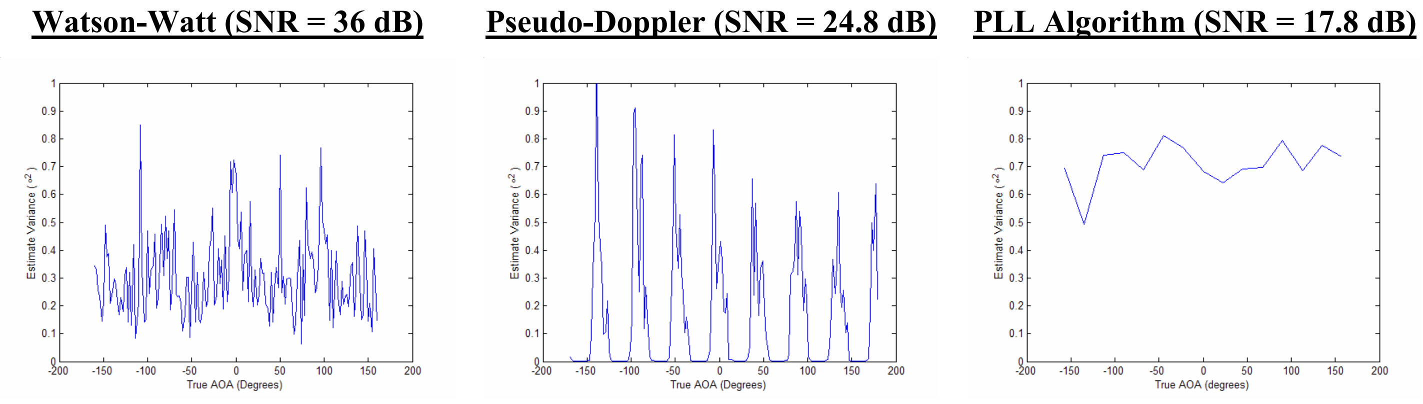


Figure 4.13 Comparisons of the SNRs of the Three Algorithms for Variance < 1^o

When looking at the Pseudo-Doppler plot, the spikes occur in the variance at the location of the antenna elements for an eight element circular array. We can see that the PLL algorithm is roughly 18 dB better than the Watson-Watt algorithm and 7 dB better than the pseudo-Doppler algorithm without any spikes occurring at the antenna elements.

Therefore, after examining the PLL algorithm and the above algorithms, we decided implement the PLL algorithm on the DRS-SS WJ-8629A receiver due to its differences from older established analog algorithms and overall improvements in accuracy to the currently explored field of single channel direction finding algorithms.

Chapter 5

Implementation of Single Channel DF

The next part of this work was concerned with the implementation of the PLL algorithm on a software-defined radio. The DRS Signal Solutions, Incorporated (DRS-SS) WJ-8629A Software Definable Receiver with Sunrise™ Technology was provided by DRS-SS Technologies for this purpose. The antenna array and switching circuit was fabricated in the MPRG research laboratory. In this chapter, we present the implementation, including the hardware and software architectures, the test setup, and the resulting performance.

5.1 DRS-SS WJ-8629A Software Definable Receiver with Sunrise ™ Technology

The implementation of the PLL algorithm was performed on a Texas Instruments DSP-based software radio provided by DRS-SS. The main processing unit is the Texas Instruments TMS320C6701 DSP processor with a maximum computational rate of nearly 1GFlops. The WJ-8629A receiver has Frequency coverage from 20 to 2700 MHz with 10-Hz resolution, and Digital data (Audio, Video, I&Q, Magnitude & Phase) or direct A/D sample data available on VXI interface. It has 22 filter slots (200 Hz to 1.23 MHz) and 5 reserved slots for user-downloadable custom filters [4]. The radio allows one to develop algorithms for the 5 reserved slots in the C programming language or the TMS320C67x assembly language.

A block diagram of the logic flow of data is shown in Figure 5.1. The RF input signal is first processed by Block 1 where it is digitized and passed to Block 2. Block 2 converts the real digital signal to baseband in-phase and quadrature (I/Q) data.

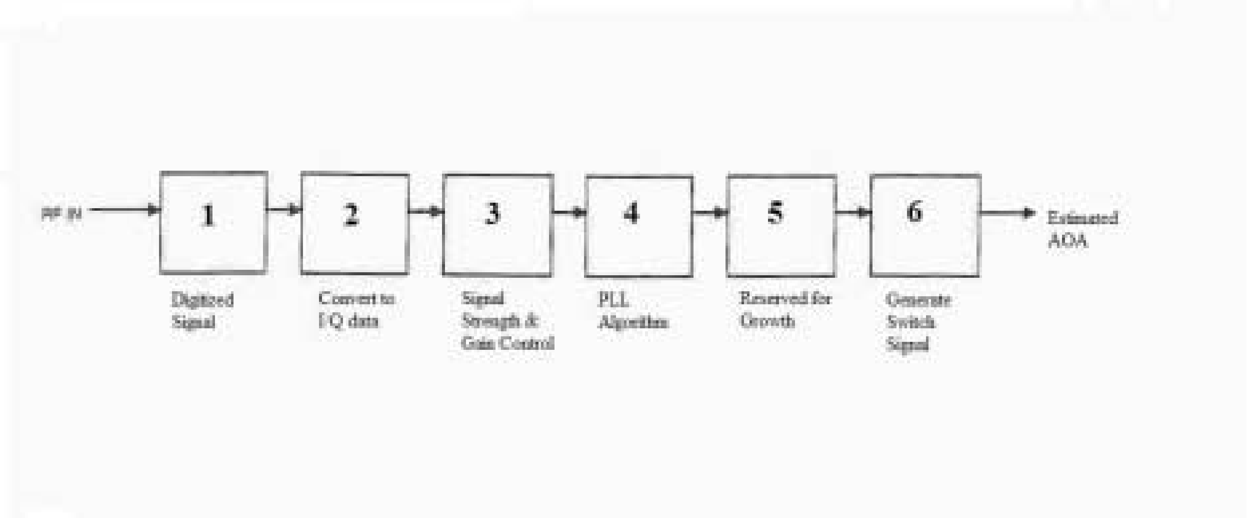


Figure 5.1 Progression of data through the software radio

Block 3 immediately calculates the signal strength of the received data and adjusts the receiver gain accordingly. Once the input data has been gain adjusted the first processing module, Block 4, is executed. The Block 4 module generally accepts a block of I/Q, magnitude, phase, and frequency data points for the PLL algorithm. Once the data has been processed by the Block 4, the data is passed to the Block 5 module as a block of I/Q data. Block 5 is reserved for later growth of the PLL algorithm Lastly, I/Q data is passed to the Block 6 module for generating the switching signal needed to progress the switch through the antenna array. Along with the switch signal, Block 6 displays the estimated AOA.

The user can develop algorithms for the demodulator, analyzer, and decoder in either the C programming language or the TMS320C67x assembly language. For our implementation, the PLL algorithm was implemented using the C programming language.

5.2 Hardware Implementation

With the components listed in Section 5.3, we can design an antenna switching circuit that is needed to pass the signal from the eight-element circular array to the software radio receiver. Figure 5.2 displays a block diagram of the antenna switching circuit and its interface to the antennas and receiver.

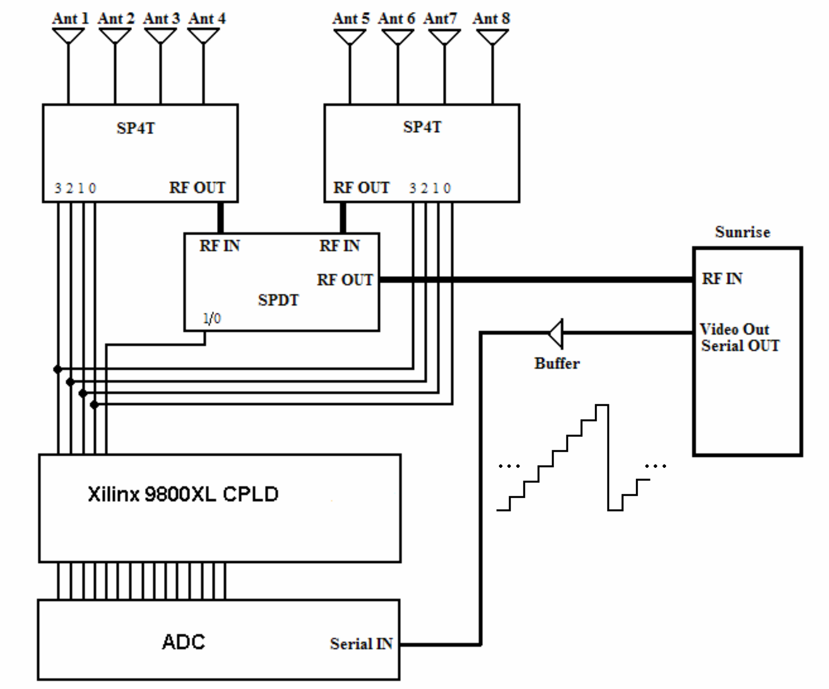


Figure 5.2 Block Diagram of the Commutative Switching Circuit

Referring to Figure 5.2, we now describe the operation of the switching circuit. First, the receiver transmits an 8-step signal to the antenna switch through a co-axial cable where each step corresponds to a different antenna. The 8-step signal is sampled and converted to a digital parallel stream that is fed into the Xilinx CPLD. The Xilinx CPLD contains code that acts like a translator. The parallel binary data entering the CPLD denotes which antenna is to be selected. The CPLD calculates the appropriate antenna based on the binary input data, and provides the TTL-driven RF switches the appropriate binary data shown in Table 5.3. The RF switches then select the appropriate antenna and the signal flows from the antenna to the receiver through one RF path.

The 8-step signal is created in such a way that the antenna array is sampled in a counter-clockwise rotation. The receiver sampling rate is set to 125 ksps and 64 samples per antenna are taken. Therefore, the circular antenna array is sampled at a rate of 244 times per second or 244 Hz array sampling.

5.3 Description of the Hardware Components

5.3.1 MPRG Antenna Array

The antenna baseline is the geometric line of interconnection between antenna elements. Antenna aperture is defined as the plane surface area near the antenna through which most of the radiation flows. The spacing between antenna elements usually determines the aperture of an array, and since we are using circular arrays, the diameter of the entire circular array determines the array aperture [5].

In order to model the antenna array, assuming a single plane wave impinging on the array, the array manifold vector for a uniform circular array can be written as:

$$U(\theta)=\begin{bmatrix} u_1(\theta) \\ u_2(\theta) \\ \vdots \\ u_M(\theta) \end{bmatrix}=\begin{bmatrix} e^{-j2\pi(R/\lambda)\sin(\varphi)\cos(\theta-\eta_1)} \\ e^{-j2\pi(R/\lambda)\sin(\varphi)\cos(\theta-\eta_2)} \\ \vdots \\ e^{-j2\pi(R/\lambda)\sin(\varphi)\cos(\theta-\eta_M)} \end{bmatrix} \tag{5.1}$$

where R is the radius of the circular antenna array, φ is the elevation angle, θ is the angle of arrival (AOA) of the incoming plane wave, η_m is the angle of the mth antenna element in the azimuthal plane, and λ is the wavelength of the center frequency of interest [5]. For simplicity, the elevation angle is set to 90^o in order to consider azimuth angles only. Angle-of-arrival estimation does not consider the effects of different elevations in this study.

The MPRG antenna array as seen in Figure 5.3 is an eight-element antenna array with a diameter of ~19.1 cm. If we assume that the received signal is a plane wave, we would like the signal will go through one complete cycle as it moves across the array. Therefore, the frequency of the CW is defined as $f = c/\lambda$ or 1.57068 GHz.

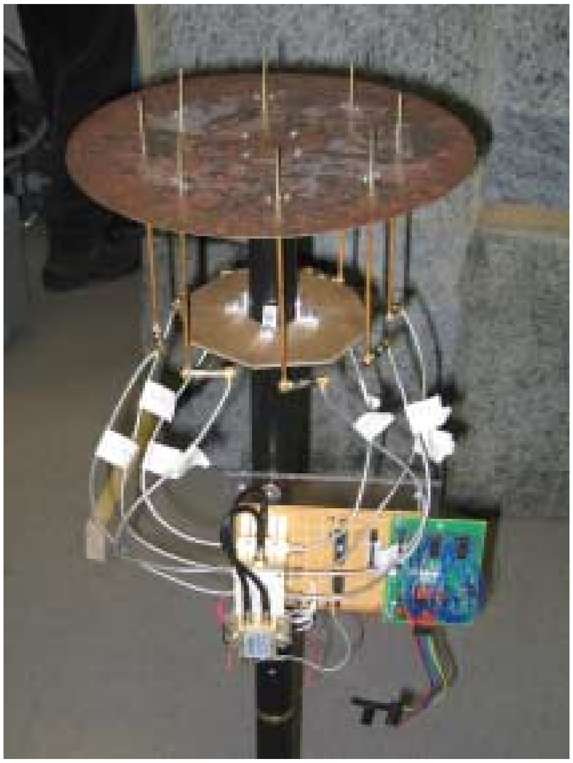


Figure 5.3 MPRG Eight Element Circular Antenna Array

In order to pass the signal from the array elements to the single channel receiver, an antenna switching circuit was designed and built as discussed in the previous section. The switching circuit had an independent power supply, but it was dependent on the receiver's switching signal to commute the switch around the circular array.

The four parts of the switching circuit are: Mini-Circuits RF GaAs Switches, a Xilinx CPLD, a TI Analog-to-Digital converter, and TI logic buffers (3.3V inverters). A list of the parts used can be found in Table 5.1.

Table 5.1 Implementation Parts List

Manufacturer	**Part**	**Model Number**	**Quantity**	**Price**
Mini-Circuits	SP4T GaAs Switch	ZSWA-4-30DR	2	$ 119.95
Mini-Circuits	SPDT GaAs Switch	ZYSWA-2-50DR	1	$ 69.95
Texas Instruments	Analog-to-Digital Converter	ADS7805	1	Free
Texas Instruments	Inverters		4	Free
Xilinx	CPLD	9800XL	1	$ 79.95

5.3.2 Mini-Circuits GaAs RF Switches

The GaAs RF absorptive switches from Mini-Circuits are rated for frequencies that range from 500MHz to 2000MHz. The specifications for the two RF switches that were used in the switching circuit are shown in Table 5.2.

Table 5.2 Mini-Circuits RF Switch Specifications

Name	Model No.	Freq. (GHz)	Insertion Loss (dB)	1 dB Compression (dBm)	In-Out Isolation (dB)
			Typ. Max.	Typ. Max.	Typ. Max.
SPDT	ZYSWA-2-50DR B	0.5 – 2.0	1.4 1.9	20 17	31 27
SP4T	ZSWA-4-30DR B	0.5 – 2.0	1.5 3.0	28 -	37 32

The RF switches are controlled using basic TTL digital logic. The TTL control logic used to select certain RF outputs (antennas in our case) can be seen in Table 5.3. Each entry marked “Low” corresponds to +0V and each entry marked “High” corresponds to +5V.

Table 5.3 Mini-Circuits RF Switch TTL Control Logic

Name	**Mod.**	**Control Ports**				**RF Outputs**			
		1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
SP4T	ZSWA-4-30DR	Low	High	High	High	On	Off	Off	Off
		High	Low	High	High	Off	On	Off	Off
		High	High	Low	High	Off	Off	On	Off
		High	High	High	Low	Off	Off	Off	On
SPDT	ZYSWA-2-50DR	Low	-	-	-	On	-	-	-
		High	-	-	-	Off	-	-	-

5.3.3 Texas Instrument Analog-to-Digital Converter

The TI ADS7805 is a complete 16-bit sampling, Analog-to-Digital (A/D) converter using state-of-the-art CMOS structures. It contains a complete 16-bit, capacitor-based, Successive

Approximation Register (SAR) A/D converter with Sample-and-Hold (S/H), reference, clock, interface for microprocessor use, and 3-state output drivers [11].

The ADS7805 is specified at a 100 kHz sampling rate and ensured over the full temperature range (-25^{o}C to +85^{o}C). Laser-trimmed scaling resistors provide an industry-standard ±10V input range while the design allows operation from a single +5V supply, with power dissipation under 100 mW [11].

The circuit that was used to convert the signal from the radio to the switching circuit is defined in Figure 5.4. The ADC was used to convert the analog switching signal from the receiver to a digital signal that could be processed by the CPLD.

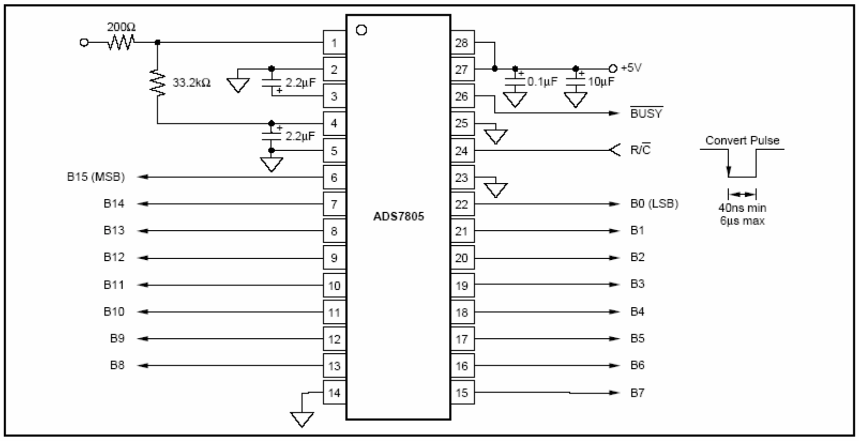


Figure 5.4 ADC Circuit Used for Parallel Operation

5.3.4 Xilinx 9800XL Complex Programmable Logic Device (CPLD)

The XC9500XL Development Kit shown in Figure 5.5 delivers an easy to use prototyping platform for the 3.3 V XC9500XL CPLD family. The XC9500XL family is targeted at high-performance, low voltage applications needing from 32 to 288 macrocells of in-system programmable logic [12].

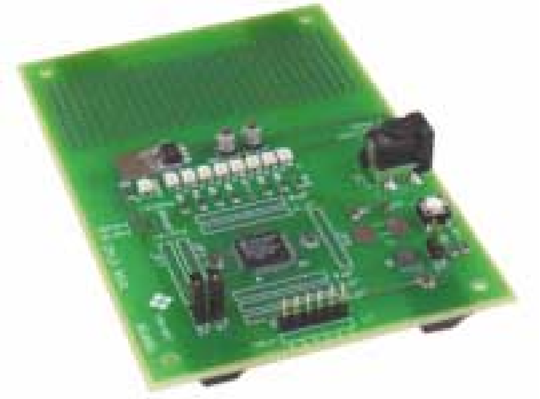


Figure 5.5 The XC9500XL Development Kit

The board included in the XC9500XL Development Kit contains the 72 macrocell, 3.3 V XC9572XL CPLD. Fifty-two user I/Os are available with the VQ64 package used. Each of these signals is brought out to header strips surrounding the CPLD, allowing for easy prototyping and expansion. A DS1073 Econoscillator provides a clock source for the board, while a TI 3.3 V regulator, Infineon LEDs, a JTAG interface and a large prototype area complete the platform design. Power can be supplied directly or through the onboard regulator. Development software to create designs for the XC9500XL family is available in the free downloadable Xilinx® ISE WebPACK™ [12]. The CPLD was used to perform the switching algorithm due to its size, speed, and price.

5.4 Description of the Software Implementation

As stated previously, the software radio receiver is sampling at rate of 125 ksps. We are taking 64 samples per antenna and sending the data to its own Phase-locked loop (PLL) as seen in Figure 5.6. Each of the eight parallel PLLs tracks the phase of the incoming signal and saves the resulting phase after the 64th sample. After inputting the data from each of the eight antennas, the receiver sends the next step in the 8-step switch signal (SS) to the antenna switch circuit to toggle to the next antenna. After the phases from all eight of the antennas have been found, the eight phases are sent to the PLL algorithm which was described in Section 3.1. After the resulting sinusoid is calculated from the PLL algorithm, the AOA estimate is created and sent to the control program.

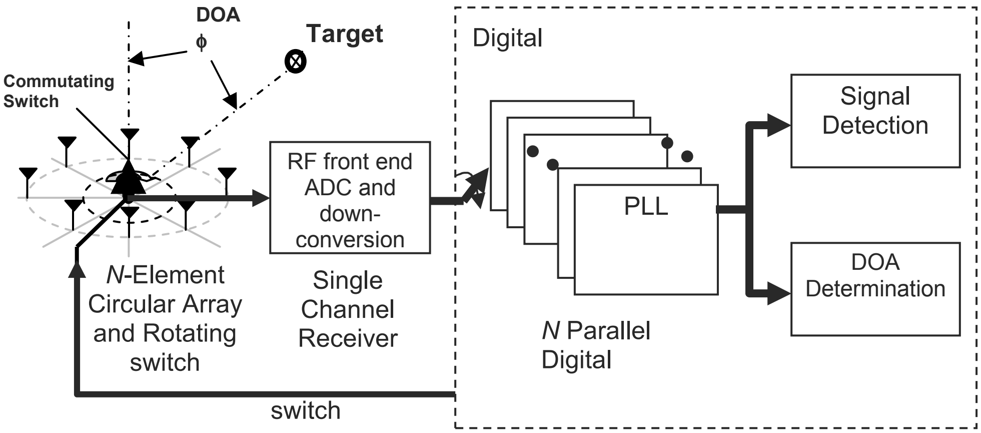


Figure 5.6 Block Diagram of the Single Channel DF System

Because the software radio receiver has limited resources within the DSP such as DSP cycle counts and memory size, we had to partition the code into different sections to run during different DSP loops.

For instance, there are not eight parallel PLLs. There is one software-implemented PLL that receives the signal from the antenna array. Because the receiver controls the array, it knows when a new antenna signal is entering the system. It also knows when it has tracked the phase of a particular antenna, and when to save it. Once it has tracked the phases of the antenna, it stores it in memory and resets the PLL.

The phases are then passed to the first part of the curve-fitting algorithm, and the PLL portion of the code is “turned off” or is no longer running. This part of the algorithm creates the lookup table that corresponds to the $2^8 = 256$ possible curves as described in Chapter 3.

After the lookup table is generated, that portion of the code is exited. The data is then passed to the minimum squared error curve-fitting technique that actually fits the phases to one of the 256 possible curves. Once the curve has been fitted, this part of the code is exited and the data is passed to the part of the code which calculates the AOA.

The AOA is calculated by taking the fit curve found in the third section of the DSP code and taking a FFT. Once the FFT is taken, we can take the second element of the FFT and use the following equation to calculate the estimated AOA.

$$\hat{\Phi} = \angle F[1] - \frac{\pi}{2} + \frac{\pi}{N_a} \tag{5.2}$$

5.5 Test Setup and Validation

In figure 5.7, we can see a block diagram of our test setup in order to validate our PLL algorithm. The test was implemented in an MPRG laboratory at Virginia Tech. In our test, we setup the antenna array with our antenna switching circuit, the DRS-SS WJ-8629A Receiver, and our transmitter.

The transmitter that was used is a Hewlett Packard Function Generator. The RF output from the function generator was then sent to a vertically polarized directional antenna that had an operating frequency between 1-18 GHz. A 10 MHz reference was connected between the function generator and the WJ-8629A Receiver to ensure that both clocks were synchronized to prevent any frequency offset from the desired frequency of 1.5707 GHz. (The ability of the algorithm to handle frequency offset between the transmitter and receiver is a critical issue that will be addressed in future work.)

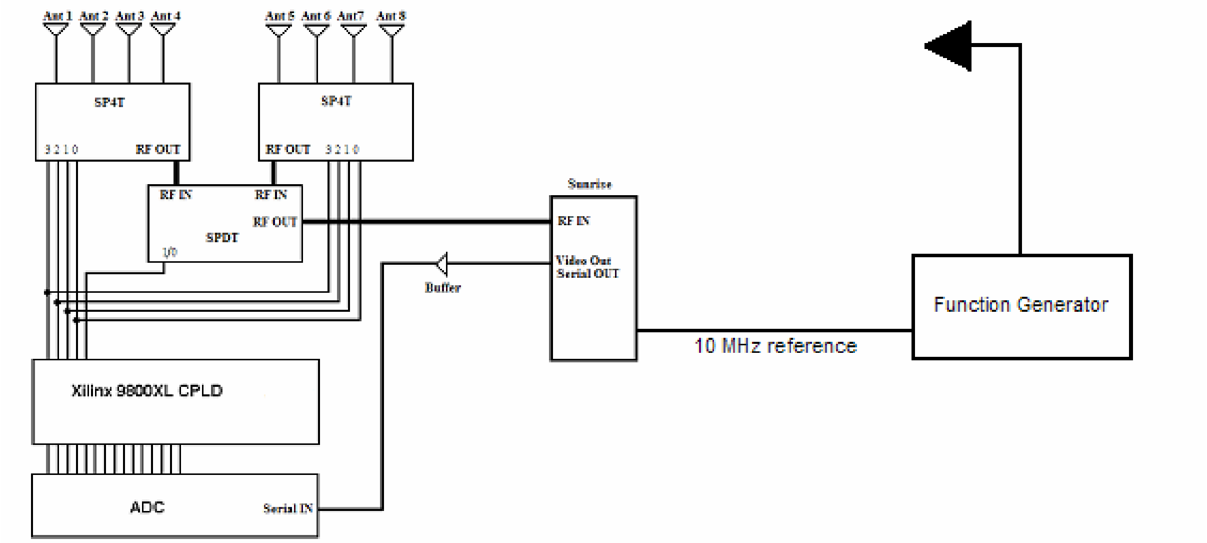


Figure 5.7 Block Diagram of the Test Setup

In order to validate the PLL method in our laboratory environment, the following screen captures were taken which show real-time processing data. We can see the switching signal that is generated by the WJ-8629A Receiver and sent to the switching circuit in Figure 5.8. The switching signal consists of 8 levels where each level corresponds to a different antenna. The first or lowest level corresponds to the East Antenna (reference 0^o). The next level corresponds to the North-East antenna. As the levels get higher, the switch progresses around the array in a counter-clockwise rotation.

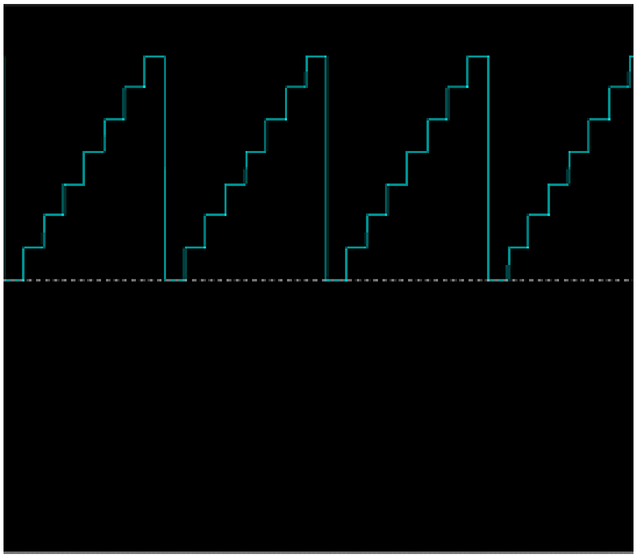


Figure 5.8 Screen Capture of the Switching Circuit Signal

In Figure 5.9 we can see the I/Q data coming into the receiver from the eight element circular antenna array. Since the receiver and transmitter frequencies are clock-synchronized using a 10 MHz signal between the function generator and the WJ-8629A receiver, there is no frequency offset. This can be seen in the fact that the I/Q values are constant over the duration of each antenna sampling.

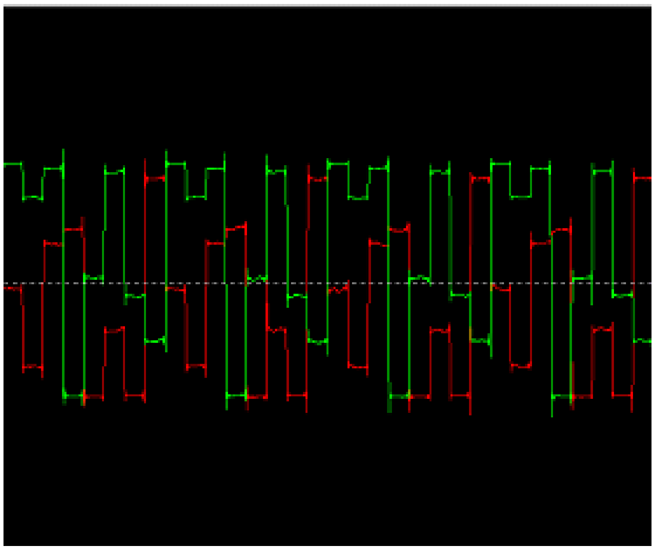


Figure 5.9 Screen Capture I/Q Data Coming into the Radio from the Antenna Array

In Figure 5.10, we plot the PLL data along with the switching circuit data in order to determine if the PLLs are converging or tracking sufficiently fast. Since we are taking 64 samples per antenna, it is very important that the PLLs converge within that timeframe. The switching signal can be lengthened if more time is needed for PLL convergence in low SNR environments. By looking at Figure 5.10, we can see that the PLLs converge after about 15-20 samples. If it were necessary to speed up our array sampling (say due to track fast moving targets), we could decrease the length of the switching signals levels to 32 samples per antenna. This would double the speed of the array sampling to 488 Hz. The control over the speed of the array sampling is beneficial in an environment which consists of frequency offsets, especially if the frequency offset is greater that 244 Hz.

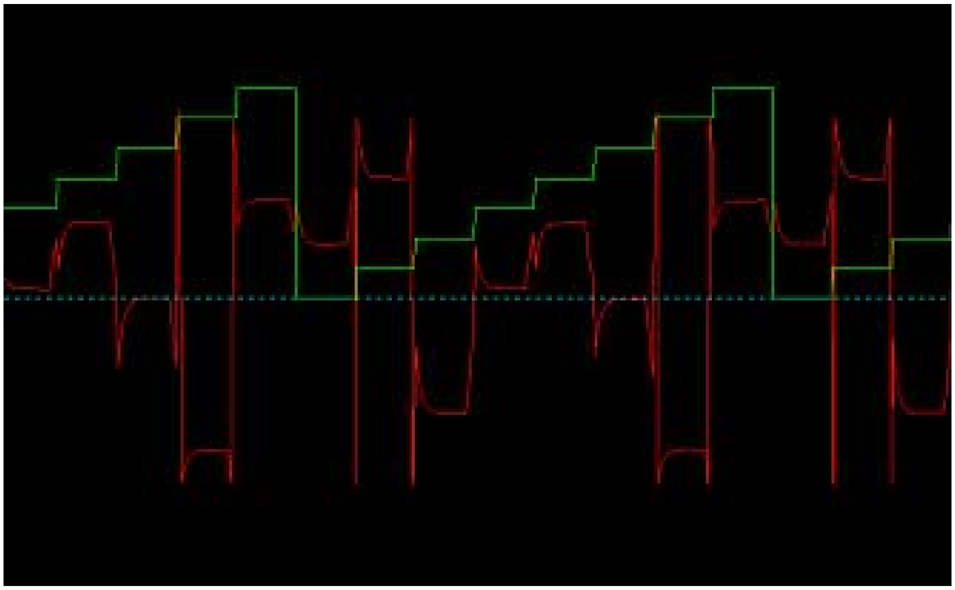


Figure 5.10 Switching Signal and the Resulting PLL for Each Antenna

5.6 Algorithm Performance

5.6.1 Performance of Algorithm with Co-axial Cable

In order to test and troubleshoot the PLL algorithm on the WJ-8629A Receiver, we first connected eight co-axial cables to each antenna input on the RF switches. Each cable was cut and phase matched in order to simulate an AOA of 0° when connected to the RF switches[1].

[1] Note that any multiple of 45° could be simulated by rotating the co-axial cable connections around the array.

After the cables were connected to the array, they were then connected to an 8-to-1 power splitter which was connected to the function generator.

Figure 5.11 plots the phases that are induced at each antenna as recorded by the receiver PLLs. We plot both the measured values as well as the theoretical values assuming no common phase offset.

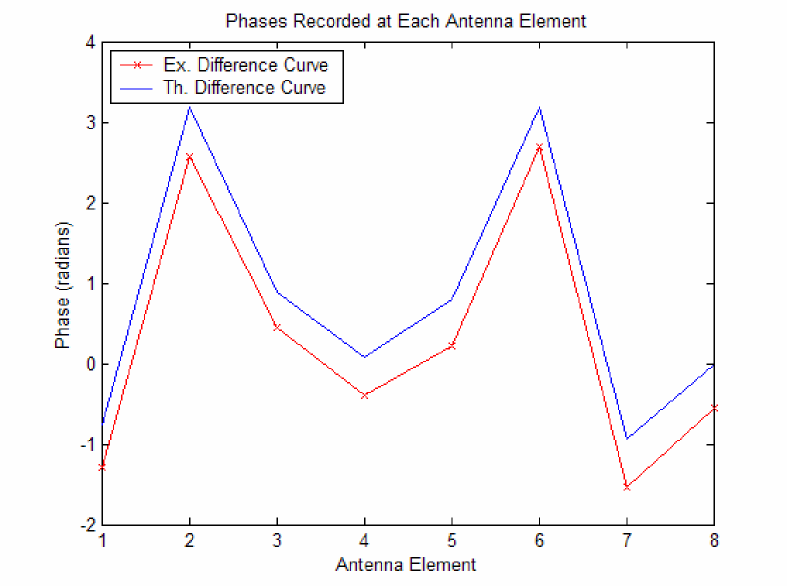


Figure 5.11 Input Phases Passed to PLL Algorithm

We can see from the above figure that experimental phases are offset slightly from the theoretical phases. This is due to a phase offset within the system. If we were to perform an AOA calculation on just the experimental phases, we would estimate the AOA incorrectly.

Now because the PLL algorithm calculates the minimum squared error on the difference curve, we can eliminate the phase offset since it is relatively constant from one antenna to the next. By examining Figure 5.12, the theoretical and experimental difference curves are nearly identical. The next step was to perform an AOA estimation using the PLL algorithm.

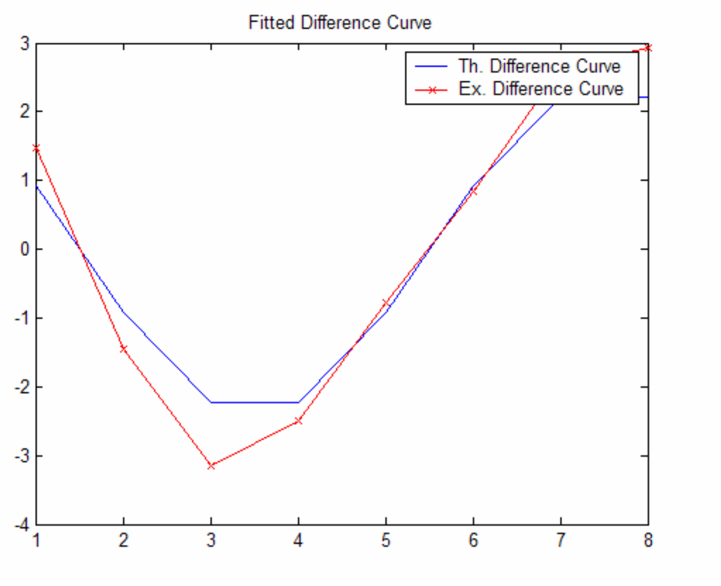
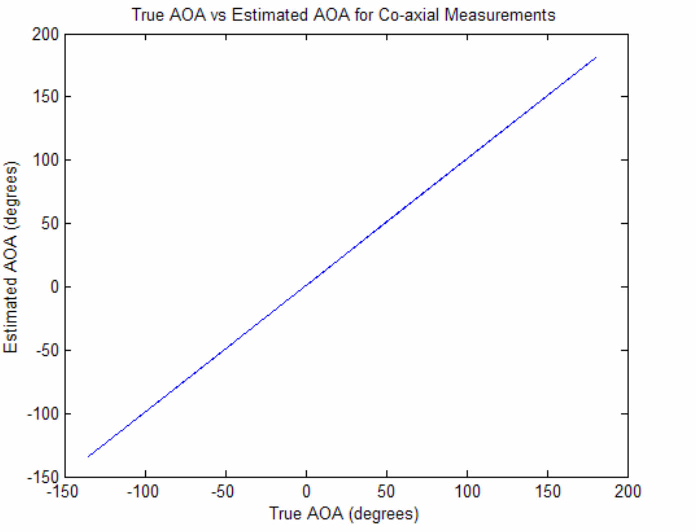


Figure 5.12 Resulting Difference Curve from PLL Algorithm

After the experimental phase values were found to be accurate, an AOA calculation was taken at for the cables configured for an angle of 0^{o}. The co-axial cables were disconnected, rotated by one cable, and reconnected to simulate an AOA of 45^{o}. This was repeated for multiples of 45^{o}. The results of the co-axial cable test are shown in Table 5.4. We see that the results are extremely accurate with a constant offset of less than one degree. We consistently get an error of 0.9389^{o} due to the slightly incorrect calibration of the cables.

Table 5.4 Co-axial Cable AOA Measurements

Theoretical AOA	Measured AOA	
0^{o}	0.9389^{o}	
45^{o}	45.9389^{o}	
90^{o}	90.9389^{o}	
135^{o}	135.9389^{o}	
180^{o}	180.9389^{o}	
225^{o}	225.9389^{o}	
-90^{o}	-89.0611^{o}	
-45^{o}	-44.0611^{o}	

5.6.2 Performance of Algorithm with Transmitter and Antenna

The previous tests confirmed that the algorithm was functioning properly and providing accurate AOA estimates. The next set of tests ran were over-the-air tests at various transmitter power levels as shown in Figure 5.13. The transmitter-to-receiver distance was 2m in a laboratory environment. The frequency was set at 1.5707 GHz and the power levels on the transmitter were varied from 10 dBm down to -80 dBm to measure the accuracy of AOA vs. Transmitter Power.

The results are given in Figure 5.13. By looking at Figure 5.13, we can see that the PLL algorithm can accurately determine the AOA of an emitter down to at least -70 dBm transmit power at a distance of 2m.

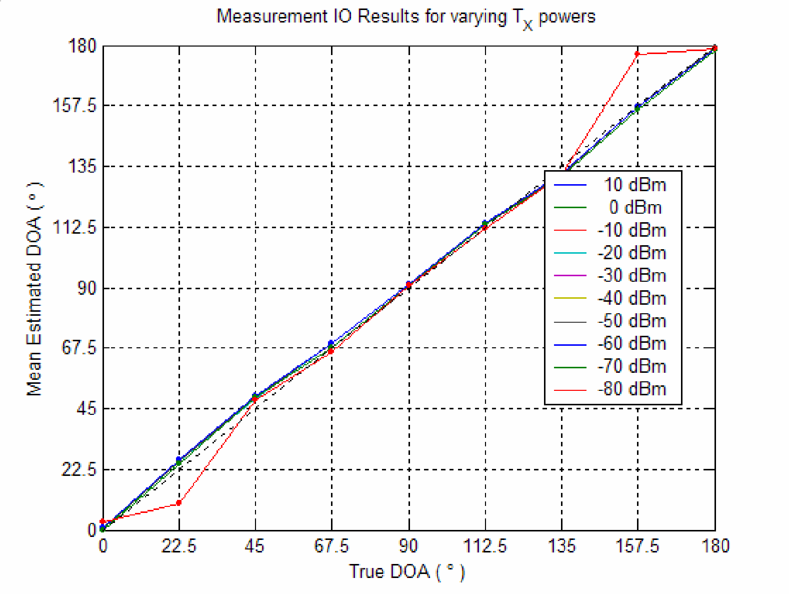


Figure 5.13 Mean Estimated AOA vs. True AOA

At a distance of 2m and a Tx power level at -80 dBm, we can see that the accuracy of the PLL algorithm starts to degrade. Because this is a digital implementation, when the performance starts to degrade, it degrades rapidly. In Figure 5.14, the error between the true AOA and the estimated AOA is shown.

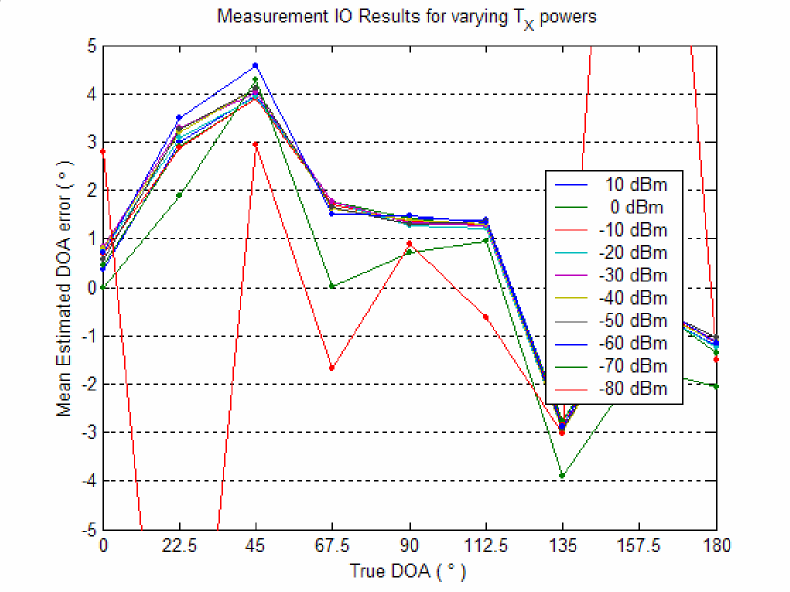


Figure 5.14 Mean Estimated AOA Error vs. True AOA

Figure 5.14 shows that there are certain AOA errors consistent at particular angles regardless of the transmitter power. Because the algorithm is unbiased when calculating the AOA, and the antenna array was not moved while the AOA was calculated; the error associated with certain AOAs is due to inaccurate knowledge of the true AOA. When testing the implementation, the true AOA was calibrated by aligning the directional antenna to the center of the antenna array by use of a laser pointer. If the laser beam was not mounted properly to the antenna tripod, then the true AOA would have a slight error. Second possible cause for the error could be found within the two SP4T RF switches. The two RF switches were not phase calibrated to each other, but each RF path within each SP4T switch were phase calibrated. In order to correct the RF phase errors without adding any additional lengths to the RF path, the Δ phase between the RF switches was corrected in the software. We either added or subtracted a correction factor for each phase that entered the receiver since the receiver has a priori information of which switch and RF path is being used. A total of nine measurements were taken between 0 and 180 degrees. We performed an array sweep at each designated transmitter power. An array sweep is defined as setting the transmitter to a certain power (i.e.: 10 dBm) and rotating the array by 22.5 degrees from 0 to 180 degrees. After the sweep is finished the transmitter power is decreased. As expected, we can see from the plot that as the Tx power is decreased, the performance of the PLL algorithm degrades. As the Tx power dips to -70 dBm and below, the performance of the algorithm starts to degrade rapidly.

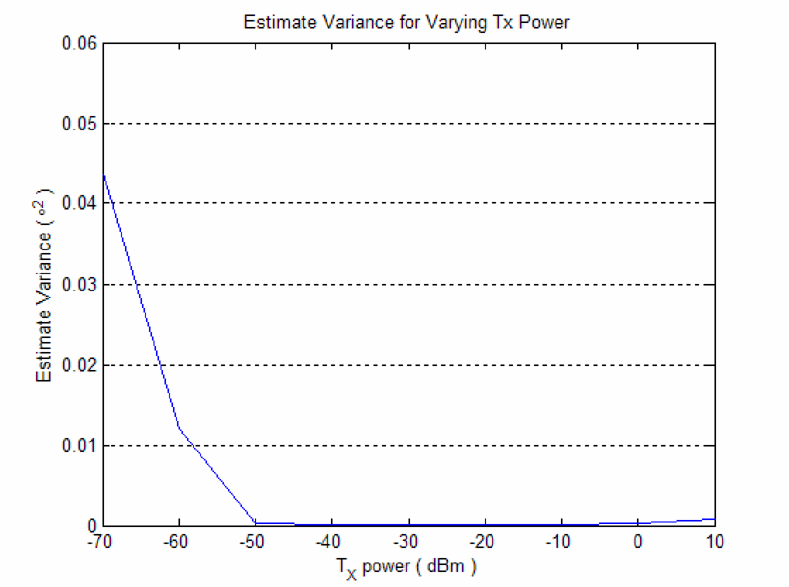


Figure 5.15 Estimated Variance vs. Transmit Power

The variance of in the AOA estimate is plotted in Figure 5.15. The variance is plotted versus transmit power which is directly related to the received power and thus SNR, We can see that the variance of the AOA starts to increase as the transmitter power decreases to below -60 dBm in Figure 5.15. Note that the error variance is actually significantly smaller than the average error. Since we are confident that the estimator is unbiased, we can be fairly confident that a large part of the average error is due to phase errors inherent in the RF switches and the inaccurate knowledge of the “true” AOA.

Chapter 6
Conclusions

The PLL algorithm has been defined and simulated in Chapter 3 and the algorithm has been implemented on the WJ-8629A receiver in Chapter 5. In this chapter we will first examine accuracy of the estimated AOA calculated in the simulations in Chapter 3 against the estimated AOA calculated on the WJ-8629A receiver as seen in Chapter 5.

Second we will discuss the obstacles that we faced and the lessons learned from these complications. We will examine the problems with the C code implementation and DSP cycle counts, the evolution of the switching circuit, and the problems experienced when setting up a valid test scenario.

Last, we will highlight ongoing and future work that will delve into the modification and optimization of the PLL algorithm and also PLL algorithm's performance in different scenarios and environments.

6.1 Comparison of Theoretical Algorithm and Implementation

In Chapter 3, we presented a number of figures that simulated the theoretical performance of the PLL algorithm in many different environments and scenarios. In Figure 6.1, we present the performance of the PLL algorithm at an SNR of 10 dB. The figure on the left plots the True AOA, the mean estimated AOA, and the estimated AOA after the PLLs have converged. We can see from the plot on the right that the AOA error is at most 1.6 degrees away from the true AOA.

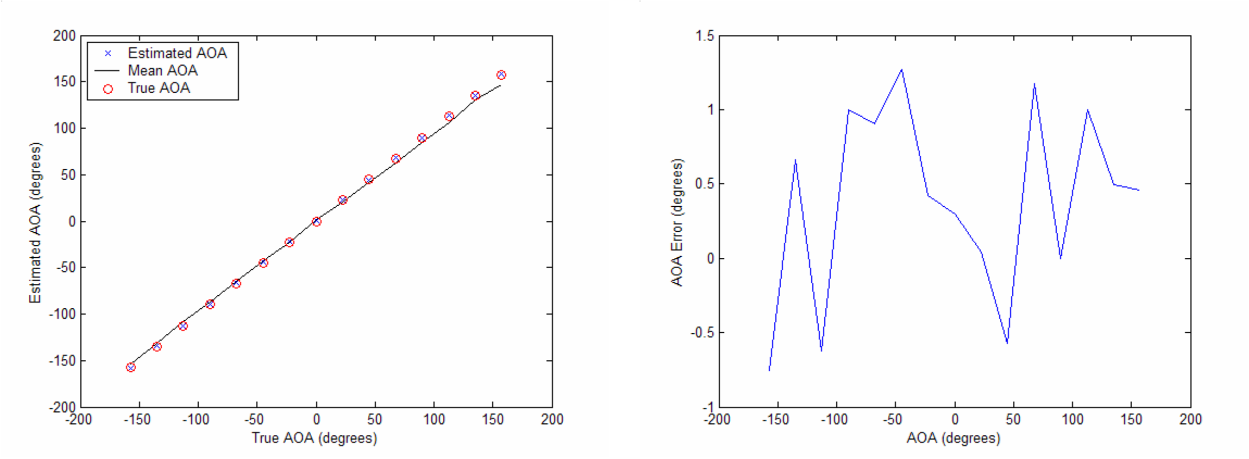


Figure 6.1 AOA Accuracy and AOA Error in an AWGN channel - Simulation (SNR = 10 dB)

Now if we examine the left plot Figure 6.2, we can see that the PLL algorithm produces an accurate estimated AOA until a transmit power of -70 dBm. At that level, the PLLs have trouble tracking the phase and the estimated AOA starts to diverge from the True AOA. We can see from the right plot in Figure 6.2 that the estimated AOA has a maximum error of ~4 degrees at a True AOA of 45 degrees when the power level is less than -60 dBm at a distance of 2 meters.

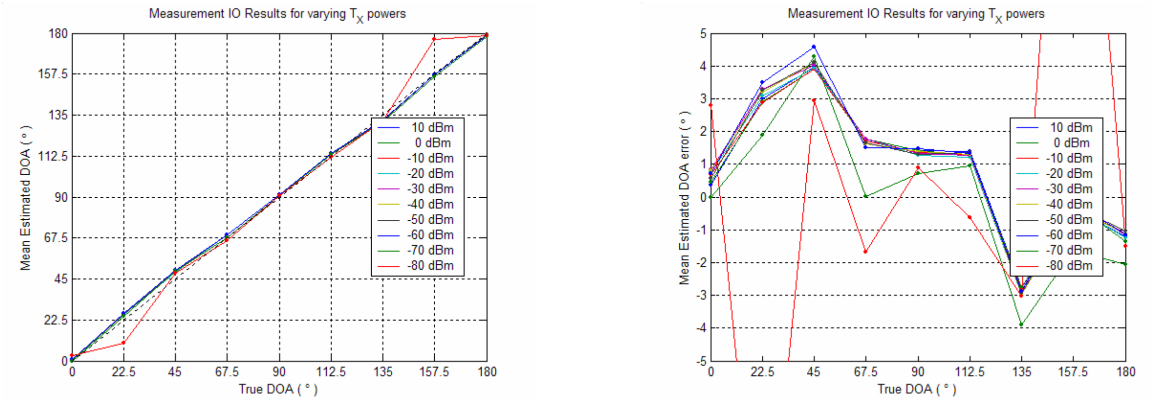


Figure 6.2 AOA Accuracy and AOA Error in an AWGN channel - Measured

Even though we cannot make a direct comparison to Figure 6.1, we can examine the maximum AOA error from Figure 6.2 at 45 degrees and compare it to Figure 6.3 which displays the mean estimated AOA after the PLL have converged for a true AOA of 45 degrees at different SNRs.

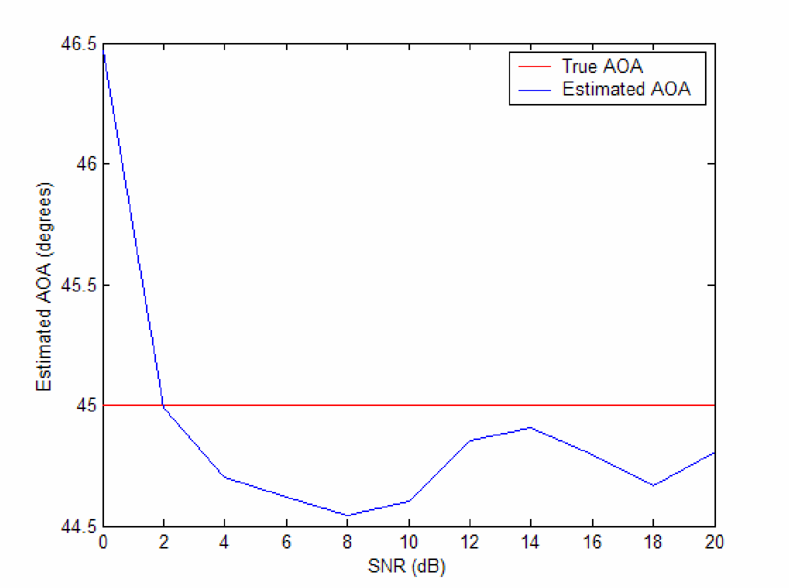


Figure 6.3 Mean AOA accuracy vs. SNR in AWGN channel - Theoretical

The maximum deviation from the true AOA in Figure 6.3 is roughly 1.5 degrees at an SNR < 2 dB. Now, Figure 6.2 shows that at different power levels that the mean AOA repeatedly has an error of ~4 degrees which is higher than what was theoretically expected after PLL convergence. This could also be due to the fact that -70 dBm corresponds to an SNR < 0 dB.

The implementation error can be caused by a number of different effects. First, we have a problem with the calibration of the True AOA. When calibrating the True AOA, a laser pointer was used to try and line up the directional transmitting antenna and the antenna array. Even though a laser pointer is more accurate than lining up the array by eye, there is still can be human error in determining the True AOA.

Another factor that led to the estimated AOA errors were the RF switches that were used. The two RF switches were not phase calibrated to each other, but each RF path within each SP4T switch were phase calibrated. In order to correct the RF phase errors without adding any additional lengths to the RF path, the Δ phase between the RF switches was corrected in software. We either added or subtracted a correction factor for each phase that entered the receiver since the receiver has a priori information of which switch and RF path is being used.

Even with these two possible causes for error, the implemented PLL algorithm can estimate the AOA to within 4 degrees. As a true AOA calibration and better correction factors are implemented for the RF switches, the AOA accuracy will improve.

6.2 Obstacles and Lessons Learned

6.2.1 Code Development and DSP cycle counts

When developing the code for the WJ-8629A receiver from our Matlab simulations, a great deal of time was consumed trying to make the code fit on the radio’s DSP. We only had a certain amount of DSP cycles per DSP loop in which all of the code had to run. With the PLL algorithm, we have to generate our lookup table, search through the table, and then calculate the estimated AOA. When we first implemented the code on the WJ-8629A receiver, we tried to implement the floating point PLL algorithm as a function in one DSP loop that would be called after the PLLs had converged. The amount of cycles that it took to run the algorithm in one DSP loop was twice as many cycles as allowed with the receiver.

In order to bring our algorithm's cycle count down into our allocated range, we had to do three things. First, we had to convert our code from a floating point algorithm in Matlab to a fixed point algorithm in the C programming language. Even though the DSP was a floating point processor, using fixed point arithmetic decreased the amount of cycles needed to run the algorithm. Second, we had to download an optimized mathematical function list from Texas Instruments which was optimized for small cycle counts. Third, we had to "break" up the code as discussed in Chapter 5. If we broke up our algorithm to run in four different DSP loops, then after the fourth loop, we would calculate an estimated AOA. When implementing the algorithm using the steps above, we were able to decrease our average cycle count down from twice the allowed cycles to within the given number of cycles which leaves room for code growth.

6.2.2 Switching Circuit

In the course of the implementation, the switching circuit that was implemented in Chapter 5 was actually the second switching circuit that was used. In the left diagram in Figure 6.4, the first circuit that was used was slower and more problematic that the one presented in Chapter 5. Instead of outputting an analog switch signal, a stream of eight data bits with start and stop bits were sent out of the WJ-8629A receiver to the switching circuit. The switching circuit's input sampling rate had to be synchronized to handle the baud rate of the receiver. After synchronization, the serial data bits were then converted to parallel data bits and latched in order to update the RF switches.

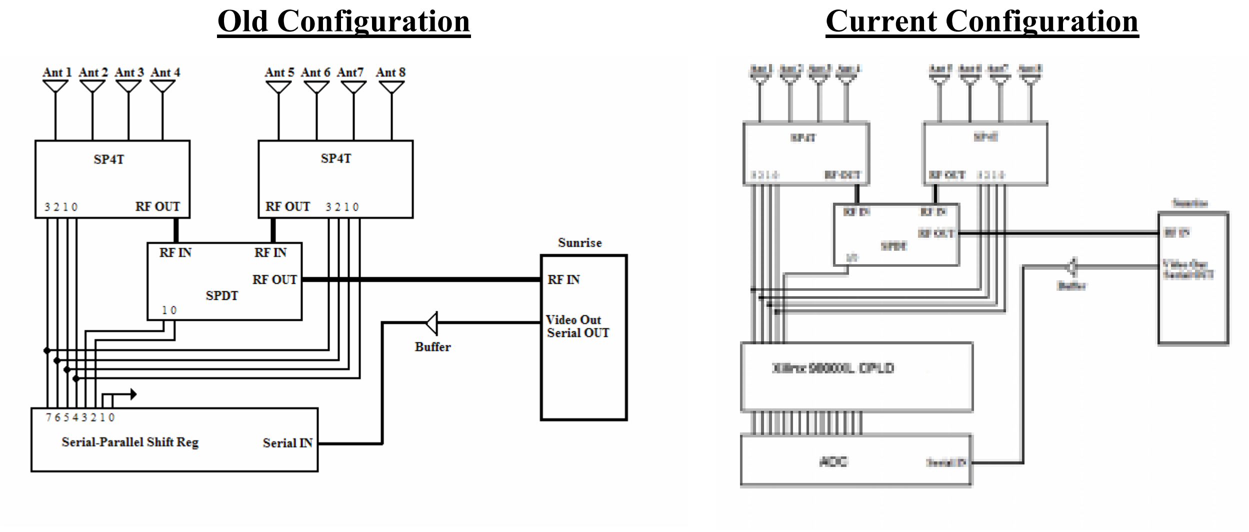


Figure 6.4 Switching Circuits Developed

The main problem with this circuit implementation was the speed at which the data was coming out of the receiver. The speed was constant and could not be increased as it can be in the current implementation. For example, in the old configuration of the switching circuit, we were only able to send out a signal to denote which switch to change every DSP loop.

In our current implementation, as seen in the right diagram in Figure 6.4, we can send out our switching signal one level at a time every DSP loop, or we can send out the entire switching signal every DSP loop. Even if we change the speed of the signal coming out of the receiver, we will not have to change clock speed of the switching circuit. This flexibility offers greater array sampling as long as the A/D converter can sample the analog signal fast enough.

6.2.3 Validation Testing and the PLL Algorithm's Second Difference Term

When we were testing our algorithm in the laboratory environment, we originally thought that the multipath presence in the lab would be too large. Most of the implementation was performed in the winter months, so packing up and going outside was not deemed desirable to try and mitigate the multipath environment.

Instead, while the weather was unpleasant, we decided to use a channel sounder to observe the different multipath effects that were present and determine where to place the transmitter and the antenna array in the lab in order to mitigate multipath. Even in the middle of the lab, we were seeing roughly 8 different significant multipath signals. Because we could not track the AOA, an action was taken to build a small anechoic chamber with some RF absorber material loaned to us from the Center for Wireless Telecommunications (CWT). Chris Anderson and I bought some lumber and constructed a small three sided chamber to place our antenna array in. When we used the channel sounder to examine the multipath in the channel, we were able to eliminate 7 out of the 8 dominant multipath effects, but our estimated AOA was still not tracking the True AOA. We then decided to take measurements outside on the Virginia Tech drill field. We set up outside and took some measurements, and we were seeing similar inaccurate AOA results as we saw in the lab. We knew then that something was wrong with our algorithm.

After examining the measured phase data in our MATLAB simulations, we were able to find the problem leading to an inaccurate estimated AOA. We noticed that the first difference curve presented in Chapter 3 was not sinusoidal and it was affecting the estimated AOA. We traced the problem from the first difference back to the phases provided by the PLLs which were imposing boundaries on the phase values coming into the receiver. This led to the implementation of the second difference calculation as presented in Chapter 5. After the second difference was implemented in the WJ-8629A receiver code, the algorithm was not only able to estimate the AOA within the homemade anechoic chamber but also without the chamber.

After the algorithm was complete, we were having trouble setting up the array in order to calibrate the "True" AOA in which to measure our algorithms performance against. As stated above, we ended up using a laser pointer to determine if the transmitting directional antenna was pointed at the correct true AOA. This is subject to human error, and the true AOA maybe +/- 1 degree away from what was setup.

6.3 Future Work and Final Thoughts

Because this algorithm is digitally implemented on the WJ-8629A software defined receiver, there are many topics that can be explored in future work by just altering the software. This is what makes this algorithm and implementation more powerful than previous single channel direction finding algorithms. We are able to modify the algorithm and upload the changes to the receiver.

The current ongoing work that is being performed focuses on how the PLL algorithm can be changed to handle a frequency offset. In our implementation, there was a 10 MHz reference to eliminate the frequency offset so that the transmitter and receiver were synchronized. If the antenna array is increased to 16 elements instead of eight, the PLL algorithm can be simplified to decrease the computational burden. Also, future work will involve listen through quality, different modulation schemes, interference cancellation, and impacts to the estimated AOA in a fading channel.

This thesis proved to be a great experience. Not only did it challenge me, but I also got to be a part of a project that started with learning the theory behind previous direction finding systems. I then worked to develop and simulate a new approach to DF systems, and finally implemented our algorithm on an actual system. This project covered all aspects of engineering design. I got to encounter the failures and successes in each part, and I feel privileged to be able to go through this experience.

References

[1] R.O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *Proc. of RADC Spectrum Est. Workshop*, Griffiss AFB, NY, pp.243-258, 1979.

[2] R.O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation" *Proc. IEEE Trans. On Antennas and Propagation,* Vol. AP-34, No.3, Mar. 1986.

[3] R.C. Johnson, *Antenna Engineering Handbook*, New York, NY: McGraw-Hill, Inc., 1993.

[4] "WJ-8629A VXI Software Definable VHF/UHF VXIceptor", DRS Technologies, Inc., Gaithersburg, Maryland. Online February 2005. http://www.drs.com/products/index.cfm?gID=17&productID=620

[5] J.A. Tsai, R.M. Buehrer, B.D. Woerner, "Spatial Fading Correlation Function of Circular Antenna Arrays With Laplacian Energy Distribution", *IEEE Communications Letters*, VOL. 6, No. 5, May 2002.

[6] R. Hammerle, "Factors Limiting the Accuracy of Doppler and Adcock Direction Finding Systems", Proc. of 1989 IEE Colloquium on Passive Direction Finding, pp.3/1-3/13, Jan. 1989.

[7] D. Peavey and T. Ogunfunmi, "The Single Channel Interferometer Using a Pseudo-Doppler Direction Finding System", *Proc. of 1997 IEEE Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, pp. 4129 – 4132, April 1997.

[8] RF Products, *A Comparison of the Watson-Watt and Pseudo-Doppler DF Techniques*, Web Note WN-004, RF Products, April 1999.

[9] N. Cianos, "Low-Cost, High-Performance DF and Intercept Systems", *Proc. of 1993 WESCON Conference*, pp. 372-376, September 1993.

[10] T.S. Rappaport, *Wireless Communications: Principles and Practice 2nd Edition*, Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[11] "ADS7805, 16-Bit 10us Sampling CMOS Analog-to-Digital Converter", Texas Instruments, Dallas, Texas. Online September 2004. http://focus.ti.com/docs/prod/folders/print/ads7805.html

[12] "Xilinx Development Kits - DS-KIT-95XL", Memec, San Diego, California. Online Product Brief September 2004. http://legacy.memec.com/devkits/americas.shtml

Vita

John Keaveny was born in St. Louis, Missouri on November 25, 1977. He attended St. Louis University High School and later enrolled at Parks College of Engineering and Aviation at St. Louis University. While at St. Louis University, he pursued a Bachelor of Science degree in Electrical Engineering.

After graduating from St. Louis University, John enrolled at Virginia Polytechnic Institute and State University in the Fall of 2002. While pursuing a Masters degree in Electrical Engineering at Virginia Tech, he joined the distinguished research group: Mobile and Portable Radio Research Group (MPRG).

While at MPRG, his work focused mainly on implementing adaptive modulation techniques and direction finding algorithms on software defined radios.

John's research interests include wireless communications, software defined radios, error correcting codes, direction finding algorithms, interference cancellation, antenna theory, and simulation of communication theory and systems.

His hobbies include visiting museums, golfing, biking, hiking, gardening, the St. Louis Irish club, and following St. Louis Cardinal Baseball.

John Keaveny now lives in St. Louis with his lovely wife Leslie. He recently accepted a full time position at the Boeing Company in St. Louis where his work focuses on analysis of RF propagation, satellite communication links, interference cancellation, antenna performance, direction finding algorithms, and radar systems.