MULTI-CHANNEL COHERENT RADIO RECEIVER AND DIRECTION FINDER FOR TRANSIENT SIGNALS

BY
TODD ARTHUR BORROWMAN
B.S., University of Illinois at Urbana-Champaign, 2004

THESIS
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2007

Urbana, Illinois

Acknowlegements

Thanks, for assistance in many ways, are due to Dr. Larry Pater, Dr. Michael White and Dr. Timothy Hayden of CERL, to Dr. Ronald Larkin of the Illinois Natural History Survey, and to Prof. Martin Wikelski of Princeton University. For their help in the building and testing of the receiver and direction finder, thanks go to Jeff Mifflin, Robert Kline, Tim Eggerding, Tim Onder and Ben Kamen. And thanks to Prof. Steven Franke and Prof. George Swenson, Jr. for their guidance and advice.

This work was supported financially by CESU Agreement No. W9132T-06-2-0011 between the U.S. Army Engineer Research and Developement Center, Construction Engineering Research Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Prof. Steven Franke and Prof. George Swenson, Jr. are the principle investigators of the project.

Contents

1 Introduction

Direction finding systems have been a great help to the environmental sciences by enabling the tracking of wildlife through radio transmittters. Originally the bearings were found with hand-held antennas, either a highly directional antenna, or one with a narrow null in the pattern. As the tracking of animals has gained importance, automated systems for direction finding and animal tracking have been developed. These systems allow more information on the animal's location to be recorded continuously and quickly. However these direction finders are an automation of the hand-held techniques and do not take advantage of the hardware and computational power now available.

The currently deployed automated direction finding system[1] uses several directional antennas arrayed in a circle. The receiver for this system is switched sequentially between successive antennas and a bearing is calculated once all the antennas have been read. The bearing of a received pulse is calculated by comparing the amplitudes of the received pulses from the two adjacent antennas with the strongest signals. The ratio of the largest to second largest signals gives the bearing with respect to the boresight direction of the antenna with the strongest signal. The signals received by the other antennas are ignored.

Also in the current system the receiver must tune for each different transmitter frequency separately, reducing the number of bearings for each transmitter since the time spent receiving must be shared between all transmitters. This need to retune also requires more complicated hardware in the receiver. The last filtering stage must have a small enough bandwidth to eliminate the other transmitters at frequencies near the desired transmitter and the oscillators must be precise enough to place the desired transmitter's image in that small bandwidth. Another drawback of this receiver is that it only records the amplitude information from the antenna, ignoring more data that could be utilized for

direction finding.

The multi-channel coherent radio receiver and direction finder uses a different approach. The antenna is less complicated, using simple dipoles instead of Yagi-Uda directional antennas. This less directional design allows every antenna to receive each pulse and use all the information collected in the bearing calculation. The mutual impedance of the dipoles is used in the design to increase the azimuthal differentiation. In similar systems that have been studied, this mutual impedance is seen as causing a perturbation from the assumed pattern and must be corrected to obtain true bearings[2].

The use of a software defined receiver increases the speed and flexibility of the system. The receiver is able to record a large band of frequencies at once and then to perform the bearing calculation on each signal received in that band. Each antenna is connected to an independent quadrature receiver so the signals are received simultaneously on every antenna. This increases the speed of the direction finder by receiving multiple transmitted signals on every antenna at once. Both in-phase and quadrature signals are used in the bearing calculation, increasing the information collected per signal, thus increasing the accuracy of the estimated bearing[3].

The new direction finding system has been tested incrementally through to the final completed system. The antenna was modeled using computer software to visualize the mutual impedance contribution to the antenna patterns and the difference from idealized models. These calculated patterns were then used in direction finding simulations to test the viability of the designed antenna and direction finding algorithms. The antenna models were also verified independently by field tests which measured the antenna patterns. Finally the entire system was tested in the field using actual animal transmitters and GPS coordinates as test data.

2 Direction Finding System Overview

The direction finding system is comprised of four parts: antenna, RF front-end, software-defined receiver, and detection and direction finding algorithm. Each component is independent, completing its task entirely before the data is sent on to the next part. This made for quicker, parallel development. Also if different antennas or a change to a different frequency band is desired only certain components need to be replaced while the vast majority of parts remain the same.

2.1 Antenna

The antenna is a four element dipole array (Fig. 2.1.1). The aluminium dipoles are 85.7 cm long and each dipole is attached to the ends of adjustable PVC arms set in a cross pattern. The arms can telescope between 58 and 68 cm and can turn 360 degrees to receive various polarizations (Fig. 2.1.2). For the field experiments the arms were set to 63.8 cm ($.35\lambda$ at 164.5 MHz).

Each dipole is center-fed with 75 Ohm RG6 coaxial cable connecting each dipole to an independent receiver channel. Chokes are attached to these lines to suppress antenna currents on the outer conductors of the transmission lines. A pre-amplifier was not used at the antenna in the current experimental setup but could be added to eliminate signal loss caused by the attenuation in the line. Alternatively, the receiver could be placed at the antenna to avoid long transmission lines and thus loss.

2.2 RF Front-End

The antenna lines are fed to the two-stage RF front-end (Fig. 2.2.3). It has four independent channels, each consisting of a preamp and two down-converting and filtering stages. The preamp is a low noise preamp with 10-11 dB of gain. The

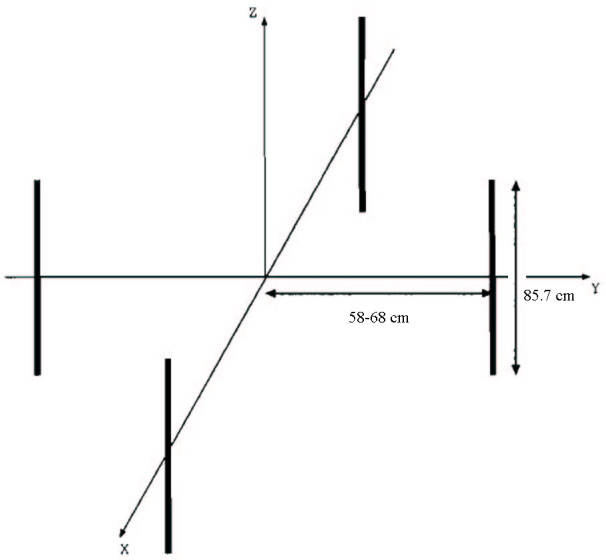


Figure 2.1.1: Layout of 4-element dipole array.

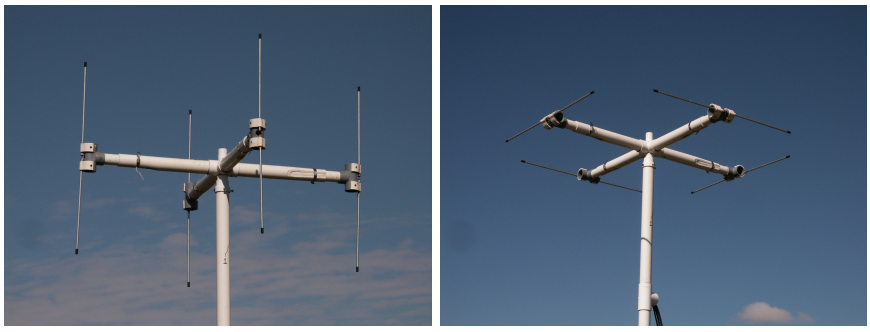


Figure 2.1.2: Antenna in the vertical (left) and horizontal (right) configurations.

signal is then mixed with the output from an external oscillator. This oscillator is tuned to $f_{Transmitter} - 70$ MHz so that the output of the mixer is at a frequency of 70 MHz. The oscillator is passively split to each channel so as to maintain the correct phase relation between channels. An internal computer-controlled oscillator will replace this external oscillator in a future version.

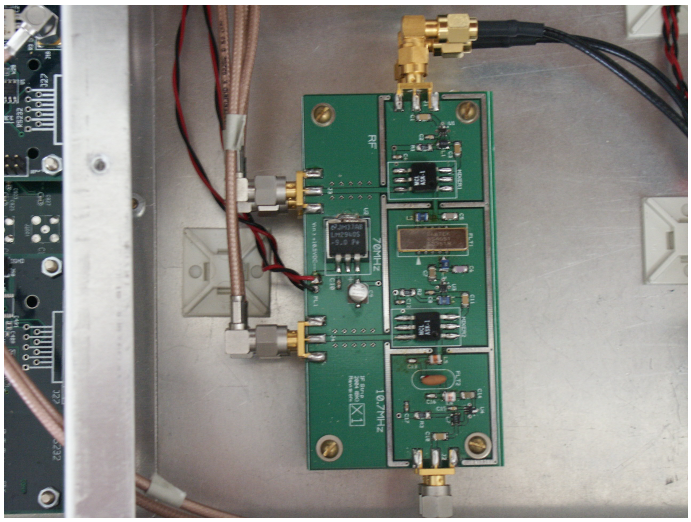


Figure 2.2.3: RF Front-End including input signals from the pre-amplifier (top), external oscillators (left), and output to USRP (bottom).

The 70 MHz output of the mixer is filtered, to eliminate duplicate images due to the mixing, with a surface acoustic wave (SAW) filter with center frequency 70 MHz and 500 kHz bandwidth. At this point the signal enters an amplifier with 22 dB of gain to recover losses incurred in the mixing and filtering. The second stage uses a fixed oscillator at 80.7 MHz to mix the signal down to 10.7 MHz. The signal is filtered using a ceramic filter with center frequency 10.7 MHz and bandwidth of 200 kHz. Finally the signal is sent to another amplifier with 22 dB of gain to recover losses.

2.3 Receiver

The receiver itself consists of computer controlled operations. The output of the RF front-end is fed to a Universal Software Radio Peripheral (USRP, Fig. 2.3.4) [4]. The input to the USRP is amplified using a programmable gain amplifier (PGA) with gain 0-20 dB. The signal is then sampled with an analog-to-digital converter (ADC) giving 12 bit samples at 64 Msamples/sec. This digital signal is down-converted to 256 ksamples/sec using an Altera Cyclone field-programmable gate-array (FPGA). At this point the 4 complex channels (in-phase and quadrature data) at 256 ksamples/sec are interlaced and streamed to a universal serial bus (USB 2.0) port.

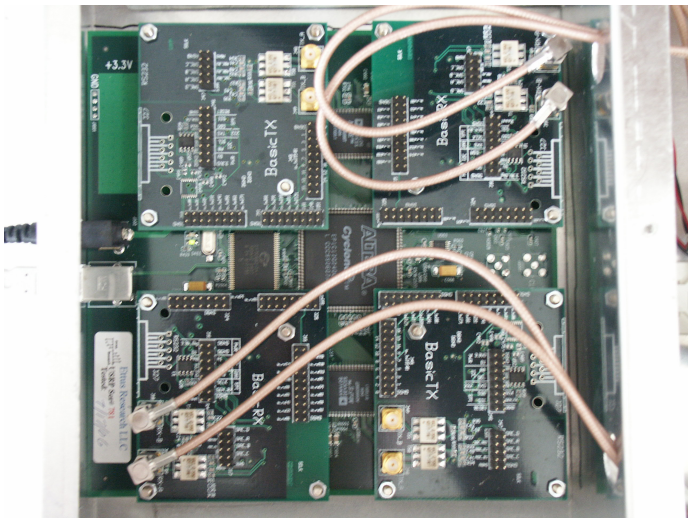


Figure 2.3.4: USRP. Each Basic-Rx daughterboard (upper-right, lower-left) receives two channels from the RF Front-End. The FPGA (center) samples the signals from the daughterboards and transmits the data via USB. The Basic-Tx modules are not used in this application.

A software-defined-radio is implemented on the computer using the GNURadio project [5]. The program reads the data from the USB port and de-interlaces the channels into separate signals. These signals are then filtered and decimated to produce 8 ksamples/sec complex signals. The center frequency of the filter

is tuned by the user, within the 256 kHz band; however, the 8 ksamples/sec is hardcoded into the reciever software. This bandwidth will be increased to allow simultaneous reception of multiple targets, but is currently limited to speed up developement and processing.

The signals are sent through an adjustable FIR filter for audio output. Also they are Fourier transformed to display the spectrum. These audio and visual cues helps the user tune the receiver. Finally signals are written to a binary file for later processing in Matlab.

2.4 Detection and Direction Finding Algorithms

The current detection and estimation analysis is implemented in Matlab for ease of development. The complex signals are read from the binary file saved by the GNURadio program (Appendix A). The amplitude of these raw signals is calculated and then filtered using a matched filter whose length is the same as the pulse width (in the field experiments the pulse transmitters had a pulse width of 20 ms). These filtered signals are then subjected to a threshold test to minimize false positives. If the filtered signals are larger than the threshold, these signals are a pulse(Fig. 2.4.5). This pulse is extracted from the data.

The Fast Fourier Transform (FFT) is performed on the detected pulse. The complex amplitudes for each channel and frequency of the pulse are calculated from the FFT by Parseval's Theorem. The complex amplitudes are then used as the input to the direction finding algorithm being implemented (See Sec. 3 for the discussion of the algorithms).

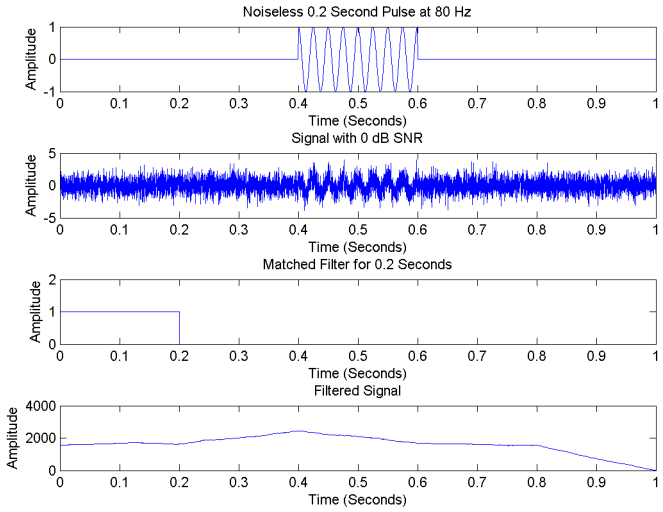


Figure 2.4.5: Pulse detection. Pulse length is 0.2 seconds in the figure for clarity.

3 Detection and Estimation Algorithms

Detection and estimation algorithms were derived and implemented in the multi-channel coherent radio reciever and direction finder. A power test is used to determine when a transmitted pulse is received. Then, depending on whether the unknown transmitter power is considered deterministic or stochastic, a direction finding algorithm is applied to the detected pulse.

3.1 Detection

A detection algorithm is used to determine the timing of the pulses received from the transmitters. This algorithm distinguishes between the pulsed transmission and noise based on the received signals. This saves processing time by not direction finding on pure noise.

The relationship between the received signals on each antenna is dependent on the direction-of-arrival. This is crucial for the direction finding algorithms but complicates the detection because the dependence is unknown. Therefore a weighted test using the covariance of the antennas is not the best solution and a simple power test was chosen.

In the detector implementation (Fig. 3.1.6, Appendix B), the power of the

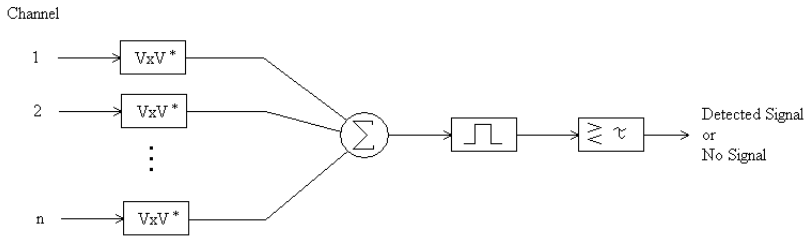


Figure 3.1.6: Flow Chart of the Detection Algorithm: Input is the n receiver channels and the output is a Boolean indicating either a Detected Signal or No Signal.

received signals is added across the channels. Then this power measurement is filtered in time with a matched filter whose width is equal to the pulse width of the transmitters (in the field tests this was 20 ms). This filtered power measurement is then compared to a threshold value (τ) and the measurements above this value are marked as detected pulses.

The threshold value is set depending on the desired percentages of false-positives (denoting a pulse where there isn't one) and misses (not detecting an actual pulse). Because the received power is dependent not only on the direction-of-arrival but also on the transmitter's distance to the antennas, the threshold should be as low as possible so as to not miss any transmitted pulses. However, the threshold should be set far enough above the noise that the detector will not be triggered on the noise; this is the cause of false-positives.

3.2 LMSE Estimation

The data model employed for the antenna is a complex vector $\mathbf{G}(\theta) = [G_1(\theta), G_2(\theta) \cdots G_n(\theta)]^T$, an n-length vector representing the antenna pattern, which is dependent on the direction-of-arrival θ. The received signals $\mathbf{V}$ are modeled as

$$\mathbf{V} = T\mathbf{G}(\theta) + \mathbf{N}, \tag{3.2.1}$$

where T is the deterministic complex transmission coefficient (based on the trasmitter distance, power and frequency) and $\mathbf{N}$ is jointly Gaussian noise with distribution $\mathcal{N}(0, \Sigma)$. This noise is assumed to be internally limited by line loss and receiver noise so the noise power is not a function of antenna gain. The conditional probability of $\mathbf{V}$ is therefore

$$p(\mathbf{V}|T, \theta) = \frac{1}{\pi^n \det(\Sigma)} \exp(-(\mathbf{V} - T\mathbf{G}(\theta))^H \Sigma^{-1} (\mathbf{V} - T\mathbf{G}(\theta))). \tag{3.2.2}$$

The maximum likelihood estimate (MLE) of T and θ are the values which maximize the conditional probability above. Bearing estimates for direction finding are produced by $\hat{\theta}(\mathbf{V})$.

$$\begin{aligned}\hat{\theta}(\mathbf{V}) &= \arg_\theta \max_{T,\theta}\ p(\mathbf{V}|T,\theta) \\ &= \arg_\theta \max_{T,\theta}\ \ln(p(\mathbf{V}|T,\theta)) \\ &= \arg_\theta \max_{T,\theta}\ -n\ln(\pi) - \ln(\det(\Sigma)) - (\mathbf{V}-T\mathbf{G}(\theta))^H\Sigma^{-1}(\mathbf{V}-T\mathbf{G}(\theta))\end{aligned} \tag{3.2.3}$$

Since the first two terms are constant with respect to T and θ and the last term is always positive the estimator is

$$\hat{\theta} = \arg_\theta \min_{T,\theta}(\mathbf{V}-T\mathbf{G}(\theta))^H\Sigma^{-1}(\mathbf{V}-T\mathbf{G}(\theta)). \tag{3.2.4}$$

If the noise is independent between channels and identically distributed (i.e. $\Sigma = \sigma^2 I$) then the noise covariance can be factored out to give the final estimator

$$\hat{\theta} = \arg_\theta \min_{T,\theta}(\mathbf{V}-T\mathbf{G}(\theta))^H(\mathbf{V}-T\mathbf{G}(\theta)). \tag{3.2.5}$$

3.3 Stochastic Case

One can change the model so that the transmission coefficient is a random variable instead of an unknown parameter. In this case T is a complex Gaussian random variable with known distribution $\mathcal{N}(0,\sigma_T^2)$. In this case $\mathbf{V} = \mathcal{N}(0,R)$ where

$$R = \sigma_T^2\mathbf{G}(\theta)\mathbf{G}(\theta)^H + \Sigma. \tag{3.3.6}$$

This changes the distribution of Eq. 3.2.2 to

$$p(\mathbf{V}|\theta) = \frac{1}{\pi^n \det(R)}\exp(-\mathbf{V}^H R^{-1}\mathbf{V}). \tag{3.3.7}$$

The MLE $\hat{\theta}$ for this model is as follows:

$$\begin{aligned}
\hat{\theta}(\mathbf{V}) &= \arg\max_\theta p(\mathbf{V}|\theta) \\
&= \arg\max_\theta \ln(p(\mathbf{V}|\theta)) \\
&= \arg\max_\theta -n\ln(\pi) - \ln(\det(R)) - \mathbf{V}^H R^{-1}\mathbf{V} \\
&= \arg\min_\theta \ln(\det(R)) + \mathbf{V}^H R^{-1}\mathbf{V}
\end{aligned} \tag{3.3.8}$$

This estimator does not simplify under independent, identically distributed (i.i.d.) noise unlike Eq. 3.2.5.

4 Antenna Modeling

The 4-element dipole antenna was modeled using computer software. These models were verified in the field by direct measurement of the antenna pattern. The computer models were used in simulations of direction finding experiments.

4.1 Nec-Win Pro

The software Nec-Win Pro [6] was used to estimate the antenna patterns for computer simulations of the direction finding. The software implements NEC (Numerical Electromagnetics Code), which uses the method-of-moments to determine the electric field produced by the antenna. The Nec-Win Pro software was chosen because it calculates and reports the electric field magnitude, phase and polarization.

Each dipole in the antenna array is modeled in nine segments (Appendices C,D,E). On the active element a 1 Volt source at 164.5 MHz is placed at the middle of the dipole. The other elements have 75 Ohm loads at their center to represent the other receiver terminations. The software generates the transmission pattern, but due to reciprocity this is equvalent to the reception pattern (Figures 4.1.7, 4.1.8, 4.1.9, and 4.1.10).

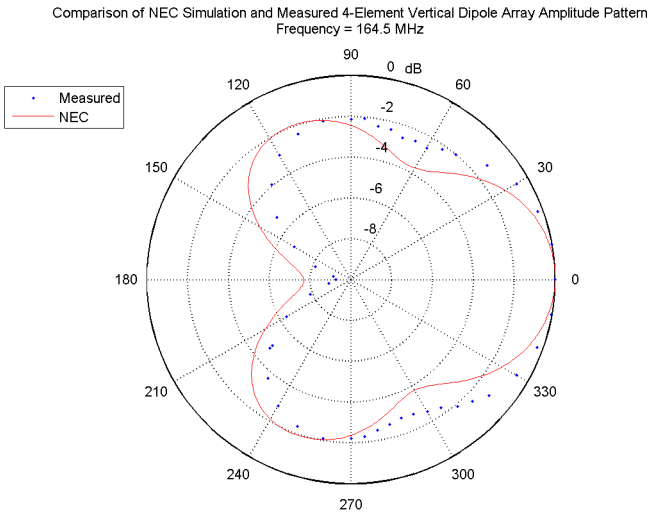


Figure 4.1.7: Vertical Polarization Magnitude Pattern

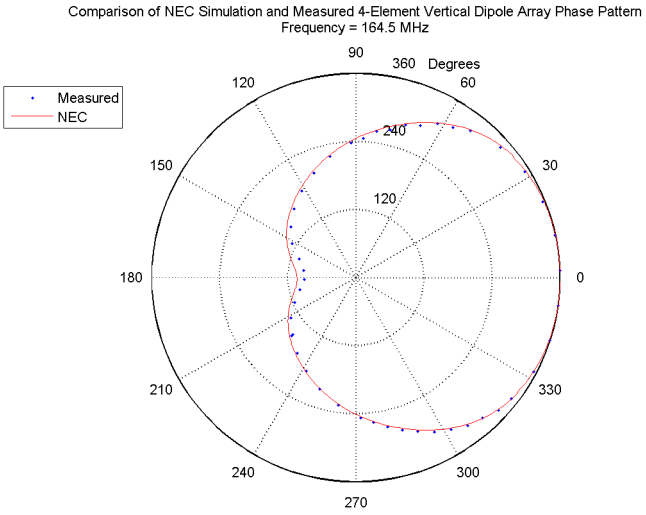


Figure 4.1.8: Vertical Polarization Phase Pattern

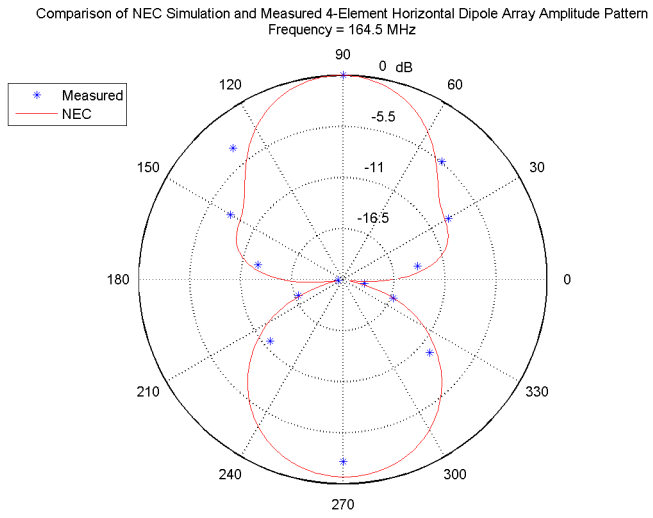


Figure 4.1.9: Horizontal Polarization Magnitude Pattern

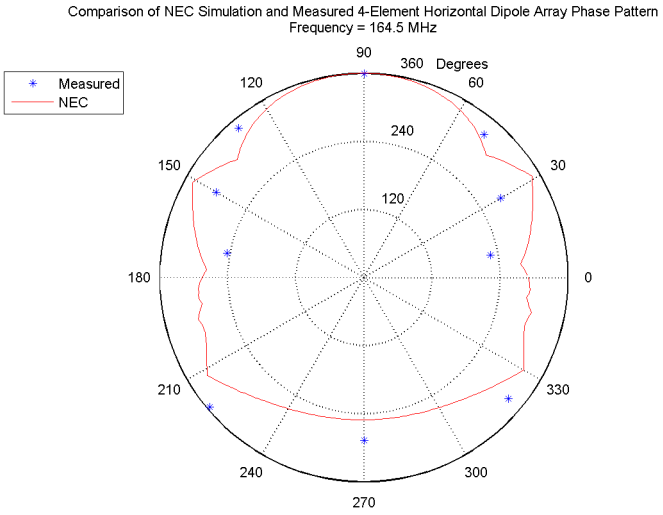


Figure 4.1.10: Horizontal Polarization Phase Pattern

4.2 Antenna Field Tests

The transmitting antenna pattern (amplitude and phase) was measured using a network analyzer (Fig. 4.2.11) at the Monticello Road Field Station of the ECE Department at UIUC. The network analyzer outputs a 50 MHz signal which is mixed with the 114.5 MHz output of the Fluke Synthesizer to give a 164.5 MHz test signal. This signal is bandpass filtered, sent through an attenuator and then to a low noise power amplifier. The amplifier output is connected to the device-under-test (the 4-element dipole array). A simple receiving antenna 24 m away (in the "far-field") receives the transmitted signal. The received signal is mixed, using the same 114.5 MHz signal, down to 50 MHz. This signal is then bandpass filtered and fed to the network analyzer.

The antenna under test was placed on a rotating platform. Twenty 1-second samples were taken at various azimuths of interest. These azimuths had a spacing of at least 20 degrees to cover the entire plane. The 20 samples at each azimuth were averaged to give the measured antenna pattern.

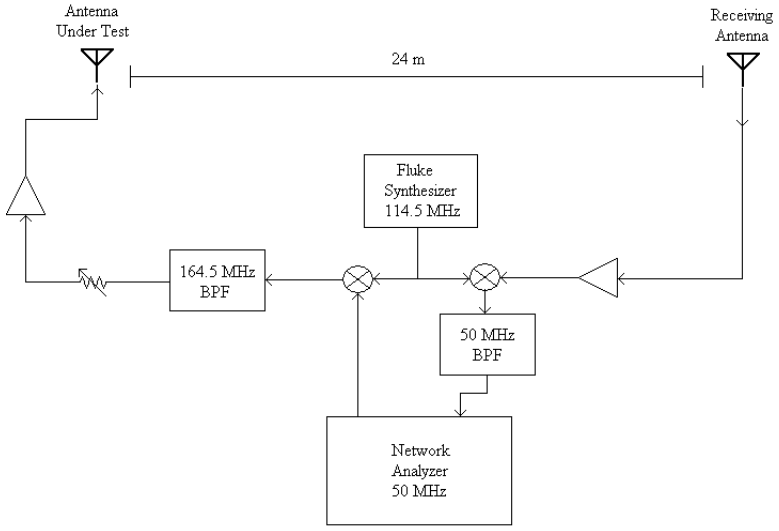


Figure 4.2.11: Network Analyzer Block Diagram

The measured vertically polarized antenna pattern has a smoother transition from the main lobe to the two side lobes as compared to the NEC simulation (Fig. 4.1.7). Also the antenna has a larger front-to-back ratio than was simulated. Both of these variations from simulation make the antenna a better direction finder by raising the gain between lobes and having an increased directional asymmetry. The phase pattern was almost identical between the NEC simulation and the measured values (Fig. 4.1.8).

For the horizontally polarized antenna pattern (Fig. 4.1.9), the measured values had a slightly broader main lobe accompanying a smaller back lobe. The phase measurements (Fig. 4.1.10) show a more elongated pattern but still matches with the expected shape.

4.3 Monte Carlo Tests

To test the usability of this antenna configuration for direction finding, a Monte Carlo test was performed using the NEC simulations as the given antenna pattern. The pattern is sampled at one degree intervals. At each sample, 100 test signals ($\mathbf{V}_i$) are generated for each signal-to-noise ratio (SNR) to be tested.

$$SNR = 10 \log_{10} \left(\frac{max(diag(\mathbf{G}(\theta)\mathbf{G}(\theta)^H))}{\sigma^2} \right) \tag{4.3.9}$$

The test signals are produced by adding Gaussian noise ($\mathbf{N}_i = \mathcal{N}(0, \sigma^2 I) + j\mathcal{N}(0, \sigma^2 I)$) to the antenna pattern, $\mathbf{G}(\theta)$.

$$\mathbf{V}_i = \mathbf{G}(\theta) + \mathbf{N}_i \tag{4.3.10}$$

These test signals are then used as the input to the estimator algorithm to be tested to produce the estimated bearing ($\hat{\theta}_i(V_i)$).

The error of the estimated bearing ($\delta\theta_i$) is calculated,

$$\delta\theta_i = \hat{\theta}_i - \theta \tag{4.3.11}$$

as is the estimator bias ($\bar{\delta\theta}$),

$$\bar{\delta\theta} = \frac{1}{100}\sum_{j=1}^{100}\delta\theta_i \tag{4.3.12}$$

and standard deviation of the error ($std(\delta\theta)$).

$$std(\delta\theta) = \frac{1}{99}\sum_{j=1}^{100}\left(\delta\theta_i - \bar{\delta\theta}\right)^2 \tag{4.3.13}$$

The bias shows the accuracy of the estimator, while the standard deviation is a measurement of precision. For optimization, one will find an antenna configuration which has the lowest maximul standard deviation in terms of frequency and azimuth.

In high SNR tests the antenna model performed well (Fig. 4.3.12). The standard deviation is relatively smooth with the larger features consistant at all the noise levels. At 40 dB SNR, all estimation errors are within one degree of the target azimuth. As the SNR decreases the standard deviation of errors increases proportionally, as is expected.

In lower SNR tests, larger estimation errors become present at around 12 dB SNR (Fig. 4.3.13). These outliers produce large spikes in the performance. These are caused by the antenna pattern's non-monotonic nature in the signal space, with respect to the parameter space. This SNR is currently below the detection threshold of the system (see Section 5.3) and thus not an immediate problem.

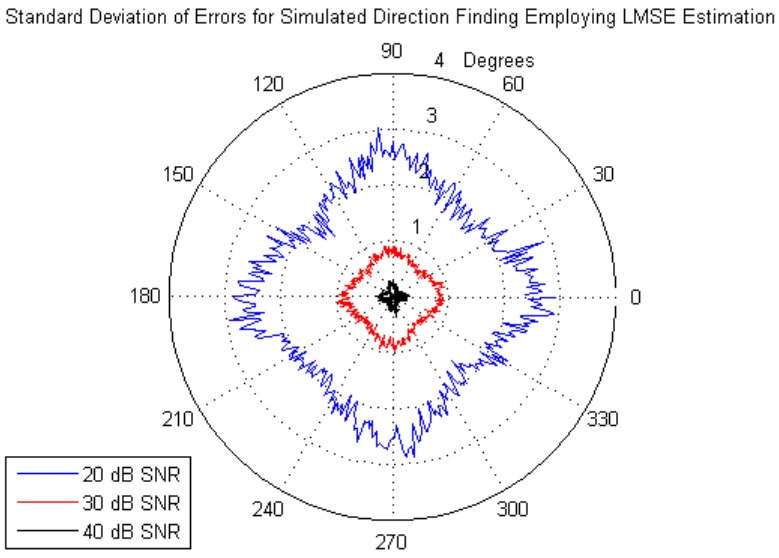
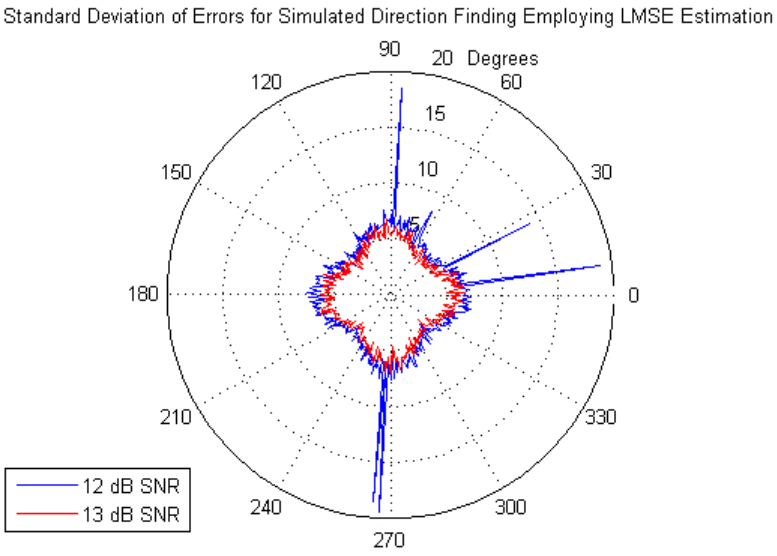


Figure 4.3.12: Direction Finding Simulation Results with High Signal-to-Noise Ratios



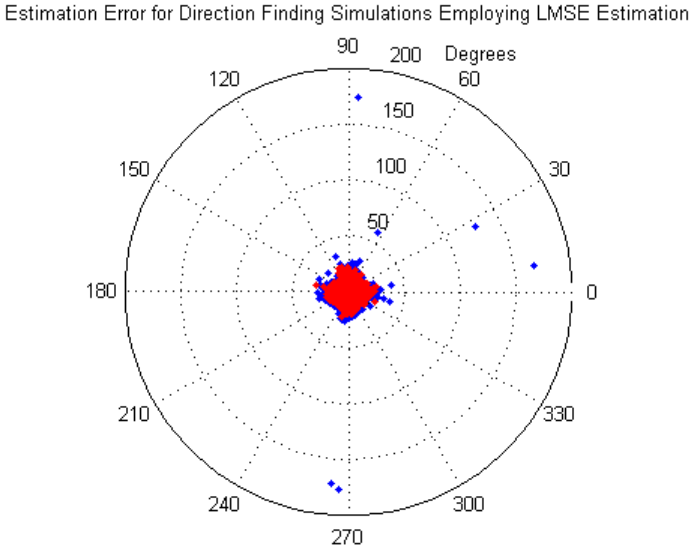


Figure 4.3.13: Direction Finding Simulations with Low Signal-To-Noise Ratios: Standard Deviation of Errors (Top) and Actual Estimates Generated (Bottom). Each point in the actual estimates is a single estimate and all estimates are from signals with independent noise. Note the outliers in the estimates cause the large standard deviation of errors.

5 Field Results

Field experiments were performed at the Monticello Road Field Station of the ECE Department at UIUC to test the equipment and verify the methods developed herein. The antenna array and receiving system was calibrated using a continuous wave transmitter. This calibration was then used to perform a proof-of-concept direction finding experiment using GPS coordinates as test data.

5.1 Antenna Calibration for Direction Finding

The antenna was mounted on a rotating platform. A continuous wave transmitter, transmitting at 164.5 MHz, was placed at the end of the gravel driveway, 370 m away from the antenna. Samples were taken for 16 seconds, every 10 degrees of rotation, using the GNURadio receiver. Post-processing was done using Matlab (Appendix F). The amplitude and phase angles were calculated for each antenna for every 10 degrees of azimuth.

In general this amplitude and phase information could be used to interpolate the antenna gain values for any azimuth. However, temporal drift in the output power of the continuous wave transmitter required signals to be normalized before the data could be interpolated to give a full, 360 degree pattern. This normalization was done by dividing each sample's four complex components by the value with the largest amplitude. This signal was chosen because it has the highest signal to noise ratio. Then these samples were interpolated using a spline interpolant to determine the pattern at 1 degree resolution. The spline interpolant is used for its ability to retain a smooth derivative.

5.2 LMSE Normalization

The LMSE needs to be changed, due to the normalization needed to be performed in the calibration. The calibration data consisted of a set of measurements that could be represented by $\mathbf{g}_i$ where

$$\mathbf{g}_i = \alpha_i \mathbf{G}(\theta_i) \tag{5.2.14}$$

and α_i is a complex, random variable. In order to be able to interpolate the calibration data to produce a pattern at any azimuth angle, the received signal needed to be normalized to remove the effect of the random parameter α_i. This was done, at each azimuth, by dividing the received signals by the strongest signal (the one with the highest SNR). In other words, a normalized array gain function was measured according to

$$\mathbf{G}_{ni} = \frac{\mathbf{g}_i}{\mathbf{g}_{i,max}} = \frac{\alpha_i \mathbf{G}(\theta_i)}{\alpha_i \mathbf{G}_{max}(\theta_i)} = \frac{\mathbf{G}(\theta_i)}{\mathbf{G}_{max}(\theta_i)} \tag{5.2.15}$$

where $\mathbf{g}_{i,max}$ is the element of $\mathbf{g}_i$ that has the largest magnitude. Likewise, $\mathbf{G}_{max}(\theta_i)$ is the element of $\mathbf{G}(\theta_i)$ having the largest magnitude. Note that normalizing in this way removes the influence of the random amplitude factor α_i. It forces the element with largest magnitude to the value 1+j0. All other elements will have magnitude less than one. After normalization, a spline fit (which produces continuous slope) is used to estimate the normalized response vectors at azimuths that were not measured. After spline interpolation, the normalized array gain function is available at any azimuth θ; we denote this function by $\mathbf{G}_n(\theta)$. When estimating the azimuth using measured data, the measured voltages also need to be normalized in the same fashion. Hence, the

measured voltages are normalized according to

$$\mathbf{V}_n = \frac{\mathbf{V}}{\mathbf{V}_{max}} \tag{5.2.16}$$

where $\mathbf{V}_{max}$ is the element of $\mathbf{V}$ with largest magnitude. The modifed function that is actually minimized in order to estimate the unknown target azimuth is thus

$$l'(\theta) = (\mathbf{V}_n - \mathbf{G}_n(\theta))^H \Sigma^{-1} (\mathbf{V}_n - \mathbf{G}_n(\theta)). \tag{5.2.17}$$

This method has been implemented in Matlab (Appendix G).

This normalization changes the properties of the noise. The known signal with additive Gaussian noise is changed into a Cauchy random variable defined as

$$\mathbf{C}(\theta) = \frac{\mathcal{N}(\mathbf{G}(\theta), \Sigma)}{\mathcal{N}(\mathbf{G}_{max}(\theta), \Sigma(max, max))}. \tag{5.2.18}$$

Since $p(\mathbf{C}|\theta) \neq p(\mathbf{G}|\theta)$, the LMSE estimator for this data scheme is not the maximum likelihood estimator. Even so the LMSE estimator works well in the noise regime encountered in the field tests.

5.3 GPS Experiment

A pulse transmitter was placed in a vehicle and driven around the Monticello Rd. Field Site. The transmitter was accompanied by a GPS receiver which recorded its location. The received signals were recorded by the GNURadio receiver. The computer was configured to receive a GPS time signal to synchronize the computer and the GPS receiver that will be with the transmitter. The GNURadio receiver was set-up with LO1 at 94.266 MHz, LO2 at 80.7 MHz and the NCO at -10.704 MHz. This placed the animal transmitter at around 500 Hz in the baseband spectrum.

The truck, with the animal transmitter and GPS receiver, initially was at

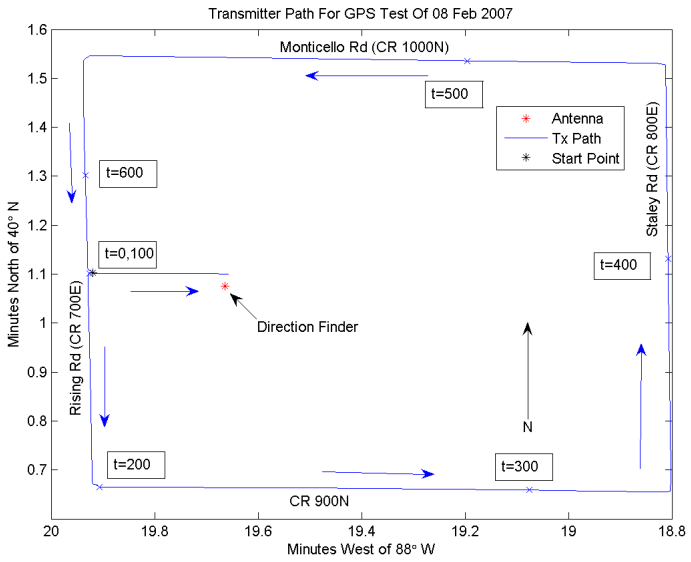


Figure 5.3.14: The Direction Finder location and the path the transmitter traversed

the end of the driveway on Rising Rd. (County Road 700E) for two minutes to line up the actual bearings and the estimated bearings (Fig. 5.3.14). The transmitting truck turned left and began traveling south on Rising Rd. At the corner of Rising Rd. and C.R. 900N the truck turned left to travel east on C.R. 900N. At the corner of C.R. 900N and Staley Rd. (C.R. 800E) the truck turned left to travel north on Staley Rd. At this point the transmitter lost the line of sight to the receiving antenna. This shows up as a lack of estimates for these bearings. At Monticello Rd. (C.R. 1000N) the truck turned left to travel west. Approximately halfway down this road the transmitter regains line of sight. At Rising Rd. the truck turned left and began traveling south. At the driveway of the field station the truck turned left and drove up the driveway east until coming to a stop at the field station building.

The bearings from the antenna to the transmitter were calculated (Fig. 5.3.15) using the GPS coordinates recorded in the truck. Then the bearings

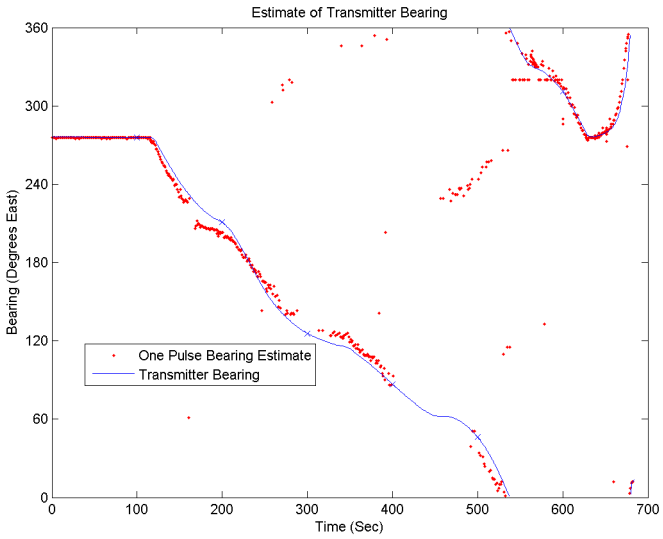


Figure 5.3.15: Transmitter bearings calculated from GPS locations and estimates. Each dot represents one pulse on which the estimate was based. The blue x's denote the bearings of the blue x's in Figure 5.3.14.

were estimated by using the LMSE estimator in Matlab (Appendix G). Each pulse detected was used independently to estimate a bearing. Signal to noise ratios (SNR) were calculated (Fig. 5.3.16) using the pulse energy divided by the noise power per Hertz. From signal S(t) the pulse energy is calculated by

$$Inst.Power = \frac{1}{R} S^*(t)S(t) \tag{5.3.19}$$

$$Energy = \frac{1}{R}\int_0^T S^*(t)S(t)\,dt = \frac{1}{rR}\sum_{k=0}^{rT} S^*(k)S(k) \tag{5.3.20}$$

where R is the resistance, r is the sampling rate and T is the length of the pulse in time. The noise power per Hertz is calculated from a noise signal n(t)

$$n(t) \longrightarrow_{DFDT} N(f)$$

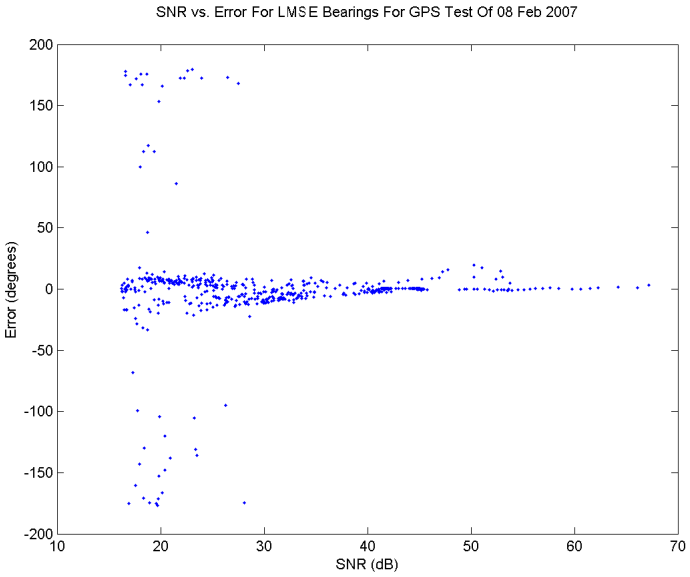


Figure 5.3.16: Error of estimate dependence on SNR. Each dot represents one pulse on which the estimate was based.

$$Power/Hz = \frac{r}{RBL} \sum_{f=0}^{\frac{LB}{r}} N^*(f)N(f) \tag{5.3.21}$$

where L is the total length of the dfdt in samples and B is the bandwidth of the noise. SNR then becomes

$$SNR = 10 \log_{10} \left(\frac{B \sum_{k=0}^{rT} S^*(k)S(k)}{r^2 L \sum_{f=0}^{\frac{LB}{r}} N^*(f)N(f)} \right). \tag{5.3.22}$$

6 Conclusion

The results of the GPS test show that the system is able to detect a single pulse and estimate bearings from that pulse. This detection and direction finding is currently done in post-processing on recorded signals. Since the receiver is software-defined it is a simple matter to do this processing in real-time as the signals are being received and thus integrating all of the processing into a single stream-lined signal processor. Due to the increased processing speed, a larger receiver bandwidth can be used, allowing multiple transmitters to be tracked simultaneously.

The antenna pattern measurements show that the NEC generated antenna patterns give good approximations for the actual physical antenna pattern. These NEC patterns can be used in direction finding simulations. These simulations, as developed, can be used to optimize the antenna array and estimation algorithms. Thus an antenna can be found that has the best theoretical direction finding attributes for this system. And from the modular design of the system the antenna can be exchanged without any changes to the hardware and only needing to update the antenna parameters in the software.

With the optimal antenna in place and using a new transmitter, the direction finder can be calibrated to use the correct version of the maximum likelihood estimator. This will improve the accuracy of the direction finder at low signal to noise ratios. The receiver will also be placed nearer to the antenna array to eliminate losses from the transmission lines, increasing the signal to noise ratio.

Improvements on the currently deployed system of direction finders for wildlife tracking have been achieved. Tool sets have been developed to take this direction finding system to the limits of theoretical precision and accuracy and with the speed and bandwidth to accurately track multiple targets in real-time.

A Matlab: Direction Finder Implementation

```
function [direction startsample frequency snr]
= directionFinder(data)
    %processes received signals, detects pulses, and
    %   estimates the direction of arrival
    %direction = estimated direction of arrival
    %startsample = sample number at the start of
    %   a detected pulse
    %frequency = frequency of detected signals
    %snr = signal-to-noise ratio of detected signals
    %data = complex signals from the receiver

    numCh=4; %number of receiver channels
    samprate=8000; %sampling rate of receiver
    threshold=6000; %threshold value for detection
    t=.02; %pulse width in seconds
    samples=size(data,2); %number of samples in the raw data
    w=floor(samprate*t); %pulse width in samples
    disp('Detection')
    %Detect signals
    [startsample]=detect(data,numCh,samprate,t,threshold);
    disp([num2str(size(startsample,2)) ' Signals Found'])
    load cAmp.mat %pulse amplitudes found by 'detect'
    load GPSVertCal.mat %antenna steering vectors
    load noise.mat %noise covariance statistics
    direction=[];
    frequency=[];
```

```
snr=[];
disp('Estimating Bearings')
%direction estimate on each found pulse
for signum = 1:size(startsample,2)
        values = lmse((complexAmp(:,signum)),(pat),n);
        [a,b]=min(values);
        direction(signum)=b;
        frequency(signum)=freq(signum);
        snr(signum,:)=((abs(complexAmp(:,signum)).^2)'./n)';
end
```

B Matlab: Detection Algorithm Implementation

```
function [startsample]=detect(raw,channels,samprate,t,threshold)
  % Detects pulses and saves the time and signal strength to
  %   file indexes.mat
  % startsample = sample where detected pulses start
  % raw = complex data from the receiver
  % channels = number of receiver channels
  % samprate = sampling rate of receiver
  % t = pulse width in seconds
  % threshold = detection threshold used to limit false
  %   positives



  checkLength=.005;% time in seconds that signal must be
        % below the threshold to denote
        % the end of a detection window
  w=floor(samprate*t);% pulse width in samples

  timefilter = ones(w,1);% matched filter in time
  sec=zeros(size(raw,1),1);
  secfiltered = zeros(size(sec,1)+w-1,1); %initialize
                                          %filtered data
  %calculate received power
  for j=1:size(raw,1)
```

```
  sec(j)=raw(j,:)*raw(j,:)';
end

%filter and compare
secfiltered = conv(sec,timefilter);
threshtest=secfiltered>threshold;

flag=0;
counter = 1;
complexAmp=[];
startsample=[];
for j=floor(checkLength*samprate):length(threshtest)
  if threshtest(j)==1 && flag==0
    startwindow = j;
    flag=1;

  elseif flag==1 &&
   sum(threshtest(j:j+floor(checkLength*samprate)))<1

    % pulse was detected
    endwindow = j;
    [maxvector, sampmax] =
     max(secfiltered(startwindow:endwindow,:));
    [maxvalue, channelmax]=max(maxvector);
    startsample(counter) =
     sampmax(channelmax)+startwindow-w;
    % isolate pulse
```

```
    pulse =
     raw(startsample(counter):startsample(counter)+w,:);
    f=fft(pulse);
    [maxvector, sampmax]=max(abs(f));
    [maxvalue, channelmax]=max(maxvector);
    %determine pulse aplitude
    complexAmp(:,counter)=
     (f(sampmax(channelmax),:).')*2/size(pulse,2);
    tempfreq = (sampmax(channelmax)-1)*samprate/w;
    freq(counter)= tempfreq-samprate*(tempfreq>samprate/2);
    flag=0;
    % correlation between channels
    co(:,:,counter) = corrcoef(pulse);
    % energy contained in pulse
    nrg(:,counter)=sum(abs(pulse).^2);
    save('cAmp','complexAmp','freq','co','nrg');
    counter=counter+1;
  end

end
numSignals=length(startsample);
save('indexes','startsample','numSignals')
```

C NEC Coding Explaination

The code which the NEC algorithm operates on defines the antenna and pattern properties one wants to simulate. The first command is the comment command or CM. Any input after this command is printed at the top of the output as a comment. The comment section is ended by the CE command. For the simple structures used in this antenna only wires were needed to be defined. They are defined by the command GW and the parameters that follow: wire number, number of segments, x-coordinate of first end, y-coordinate of the first end, z-coordinate of the first end, x-coordinate of second end, y-coordinate of the second end, z-coordinate of the second end, and the wire radius. The command GS scales the input parameters of the wire geometery by the amount in the third parameter. The command GE ends the geometery input mode with the input parameter flag denoting the presence or absence of a ground plane.

The command EX adds a excitation source on a wire segment denoted by the parameters: type of source, wire number where the source is to be located, the segment of that wire, admittance calculation and printing option, real source amplitude, and imaginary source amplitude. Loads are simulated by the command LD with the parameters: type of load, wire number for where the load is to be located, the segment number of that wire where the load starts, the segment number where the load ends, resistance of the load, the inductance of the load, and the capacitance of the load. The frequency range is defined by the command FR and its parameters: a Boolean flag specifying linear or logrithmic spacing, number of frequency steps, two blank inputs, the starting frequency in MHz, and the frequency spacing in MHz. The command RP defines the plane of the antenna pattern being calculated by the parameters: radiation pattern mode, number of elevation angles that will be calculated, number of azmuthal angles that will be calculated, flags relating to output normalization,

initial elevation angle, initial azimuthal angle, elevation incremental size, and azimuthal incremental size. Finally the pattern commands are terminated by the EN command which has no input parameters.

D NEC Code: Vertically Polarized 4-Element Dipole Array

```
CM Vertically Polarized 4 Element Dipole Array
CM frequency 164.5, theta 90, azmuthal pat.
CE
GW 1 9 0.637856 0 -.4285 0.637856 0 .4285 .01
GW 2 9 0 0.637856 -.4285 0 0.637856 .4285 .01
GW 3 9 -0.637856 0 -.4285 -0.637856 0 .4285 .01
GW 4 9 0 -0.637856 -.4285 0 -0.637856 .4285 .01
GS 0 0 1
GE 0
EX 0 1 5 0 1 0
LD 0 2 5 5 75 0 0
LD 0 3 5 5 75 0 0
LD 0 4 5 5 75 0 0
FR 0 1 0 0 164.5 1
RP 0 1 360 1000 90 0 1 1
EN
```

E NEC Code: Horizontally Polarized 4-Element Dipole Array

```
CM Horizontally Polarized 4 Element Dipole Array
CM frequency 164.5, theta 90, azmuthal pat.
CE
GW 1 9 .4285 0.637856 0 -.4285 0.637856 0 .01
GW 2 9 .4285 -0.637856 0 -.4285 -0.637856 0 .01
GW 3 9 -0.637856 -.4285 0 -0.637856 .4285 0 .01
GW 4 9 0.637856 -.4285 0 0.637856 .4285 0 .01
GS 0 0 1
GE 0
EX 0 1 5 0 1 0
LD 0 2 5 5 75 0 0
LD 0 3 5 5 75 0 0
LD 0 4 5 5 75 0 0
FR 0 1 0 0 164.5 1
RP 0 1 360 1000 90 0 1 1
EN
```

F Matlab: Antenna Pattern Calibration

```
function []=caltdat(directory)
  %Determines amplitude and phase of received signals for
  %  direction finding calibration
  %directory: directiory where the 'tdat' files are located

  numCh=4; %Number of received signal channels
  rate=8000; %Sampling rate of receiver
  cd(directory);
  files=dir;
  freqs=[];
  cAmps=[];
  counter=1;
  for j=1:length(files)
    len=length(files(j).name);
    if isdir(files(j).name)==0 && len>3
      fn=files(j).name;
      flen=length(fn);
      if fn(flen-3:flen)=='tdat' %if the file is a tdat file
        disp(fn)
        %Load tdat data into Matlab
        fid=fopen(fn,'r','ieee-le');
        raw=fread(fid,[numCh*2,inf],'float');
        fclose(fid);
        %Convert I and Q data to complex data
        craw=[];
        for k=1:numCh
```

```
        craw(:,k)=(raw(2*k-1,:)+sqrt(-1)*raw(2*k,:)).';
      end
      f=fft(craw);
      [a,b]=max(f);
      [c,d]=max(a);
      freqs(counter)=(b(d)-1)*rate/size(f,1);
      %correction so -rate/2<=freqs<rate/2
      if freqs(counter)>=rate/2
        freqs(counter) = freqs(counter)-rate;
      end
      cAmps(counter,:)=f(b(d),:)/size(f,1);
      foundfiles(counter,:)=fn;
      counter=counter+1;
    end
  end
end
%Saves complex amplitude, frequency, and file names
%   to calfiles.mat in directory
save('calfiles','cAmps','freqs','foundfiles');
```

G Matlab: LMSE Implementation

```
function value=lmse(a,p,n)
    %calculates and returns the mean squared error
    %   per azimuthal angle
    %a = received signal (4x1)
    %p = antenna steering vectors (360x4)
    %n = noise covariance matrix (4x4)



    a=a/max(a); %normalize received signals


    value=zeros(360,1);
    for j=1:360
        value(j)=(a'-conj(p(j,:)))*inv(n)*(a-(p(j,:).'));
    end
```

References

[1] W.W. Cochran, G.W. Swenson, Jr., L.L. Pater, "'Radio Direction Finding for Wildlife Research"', in *Proceedings of the 2001 Antenna Applications Symposium, Allerton Park*, 2001, pp. 295-307.

[2] B. Friedlander and A.J. Weiss, "'Direction Finding in the Presence of Mutual Coupling"', *IEEE Trans. Antennas and Propagation 39*, No. 3, pp. 273-284, 1991.

[3] G.W. Swenson, Jr., L.L. Pater and M.J. White, "'A Direction Finding System for Transient Signals"', Construction Engineering Research Laboratory, U. S. Army Corps of Engineers ERDC/CERL SR-07-3, March 2007.

[4] Ettus Research LLC, "'Ettus Research,"' http://www.ettus.com/index.html.

[5] GNURadio FSF Project, "'The GNU Software Radio,"' March 2007, http://www.gnu.org/software/gnuradio/index.html.

[6] Nittany Scientific Inc., "'NEC-Win Pro,"' 2003, http://www.nittany-scientific.com/nwp/index.htm.