UNIVERSITY OF CALIFORNIA

Los Angeles

Neural Networks for Correction of Pointing and Focal Errors on Large Deep Space Network Antennas at Ka-Band

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical Engineering

by

Ryan Mukai

2003

The dissertation of Ryan Mukai is approved.

Ali H. Sayed

Victor A. Vilnrotter

Stefano Soatto

Kung Yao

Gregory J. Pottie, Committee Chair

University of California, Los Angeles

2003

Table of Contents

Table of Figures

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Victor A. Vilnrotter of the Jet Propulsion Laboratory for his help in finding an active research project that became the topic of this dissertation, many hours of helpful discussions during the course of this work, several proofreadings of this dissertation, and for serving on the author's doctoral committee. Dr. Vilnrotter was a lead co-author of the papers listed in the vita section of this dissertation, and those works formed the foundation upon which this dissertation was built. Dr. Vilnrotter and Mr. David Girdner both spent many long days and nights conducting experiments along with the author at both DSS-14 and DSS-13, and without their efforts this dissertation and the papers in the vita would not have been possible. Mr. David Rochblatt of the Jet Propulsion Laboratory spent a great deal of time proofreading this dissertation and provided valuable advice and comments during the course of this work, including significant technical advice on the material involving RF optics and phase shifts due to deformations of the main reflector. Mr. Rochblatt's works are cited throughout this dissertation, and the 70-meter antenna holography data from Rochblatt et al. proved very valuable in the course of this research, particularly in the simulations used to develop the algorithms presented here. Dr. Gregory J. Pottie, the author's advisor, spent many hours proofreading many versions of this manuscript and made many important and helpful suggestions to improve the clarity, organization, and technical content of the text. Dr. Payman Arabshahi of JPL was a co-author of the papers listed in the vita who spent

many hours working with this author on developing RBF neural networks for pointing applications and whose many helpful suggestions led to the present work. Dr. Vahraz Jamnejad of JPL spent many hours providing helpful advice and was a co-author of some of the papers listed in the vita which form the foundation of the present work. The simulations used in the papers in the vita along with the simulations of Chapter 6 of the present work make extensive use of code written by Dr. Jamnejad. The physical optics simulations, in particular, were run on a JPL supercomputer with significant help from Dr. Jamnejad during the course of this work. The author also wishes to thank the system administrators who operate the supercomputing facility at JPL for their patience with the very large physical optics simulation jobs that occupied the processor queues on nights and weekends and his supervisor, Dr. Tsun-Yee Yan, for allowing the author time off from work to finish this dissertation.

VITA

1997	B.S., Electrical Engineering University of California Los Angeles, California
1997	M.S., Electrical Engineering University of California Los Angeles, California

PUBLICATIONS AND PRESENTATIONS

R. Mukai, V. Vilnrotter, and P. Arabshahi, "Computationally intelligent array feed tracking algorithms for large DSN antennas," *TMO Progress Report 42-141*, vol. 42-141, May 15, 2000.

R. Mukai, P. Arabshahi, and V. Vilnrotter, "An array feed radial basis function tracking system for NASA's deep space network antennas," *IEEE Joint. Conf. Neural Networks*, Como, Italy, July 24-27 2000.

R. Mukai, V. Vilnrotter, and P. Arabshahi, "Tracking performance of adaptive array feed algorithms for 70-meter DSN antennas,"*TMO Progress Report 42-143*, vol. 42-143, Nov. 15, 2000.

R. Mukai, V. Vilnrotter, P. Arabshahi, and V. Jamnejad, "Adaptive acquisition and tracking for deep space array feed antenna systems," *IEEE Transactions on Neural Networks,* vol. 13. no. 5, pp. 1149-1162, Sept. 2002.

ABSTRACT OF THE DISSERTATION

Neural Networks for Correction of Pointing and Focal Errors
on Large Deep Space Network Antennas at Ka-Band

by

Ryan Mukai
Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2003
Professor Gregory J. Pottie, Chair

Two types of neural networks, multilayer feedforward networks and radial basis function networks, are evaluated along with interpolated least squares algorithms for correcting pointing errors and focal errors on NASA's 70-meter and 34-meter Deep Space Network (DSN) antennas at Ka-band (32-GHz) frequencies. Ka-band systems offer the possibility of gains as high as 8-10 dB over existing X-band (8.4-GHz) systems, but sensitivity both to pointing errors and to deformations of the antenna's main reflector are greatly increased[7,38,52]. The correction of pointing and focal errors is thus a necessary step toward realizing the potential gains of a move to Ka-band, and algorithms for correcting antenna pointing and for correcting antenna focus via subreflector *z*-position adjustment are presented. Both simulation results and real-world experimental results show that neural networks and interpolated least squares algorithms achieve pointing and subreflector positioning accuracy that will eliminate most pointing and focal-error related SNR losses at Ka-band, significantly improving antenna efficiency.

1 Introduction

An upgrade from existing X-band (8.4-GHz carrier) systems to Ka-band (32-GHz) systems is a goal of the NASA Deep Space Network (DSN). The advantages of moving to higher carrier frequencies include an increase in antenna gains, reduced sensitivity to deep space plasma effects, and greater available bandwidth [7,52]. Such a move has the potential to yield gains as high as 8-10 dB [7,37,52].

Due to the shorter wavelengths and narrower beam width at 32-GHz, the system will exhibit increased sensitivity to both pointing errors and deformations of the ground antenna's main reflector [6,7,11,20-24,28-31,35,37-40,42,43,46-48,48-55,57]. These errors can be induced by wind, thermal gradients, errors in the pointing prediction model for a given spacecraft, and gravitational distortions of the dish itself. For example, pointing prediction models for a spacecraft, which are based upon the rotation of the Earth and upon the motion of the spacecraft relative to the Earth, sometimes contain errors of up to 10-millidegrees. This will result in severe SNR losses at the receiving horn since 3-dB losses are associated with pointing errors of 4-millidegrees. The DSN's goal is to keep RMS pointing errors within 0.8-millidegrees, an operating point which corresponds to a 0.1-dB loss of SNR[29,31]. Furthermore, the antenna's main reflector may suffer from RMS distortions of over 1.0-millimeter and peak distortions as high as 2.5-millimeters due to gravity and wind as shown in both holography-derived [37,44] and theodolite-derived [22] pictures of antenna dish deformation. Although such distortions are indicative of remarkable structural engineering of the main reflector, which has a diameter of 70-meters, the wavelength $\lambda = \frac{c}{f}$ is 9.369-millimeters at a frequency of 32-GHz. Since the effect of a distortion is doubled by reflection, a 3.0-millimeter change in RF path length (i.e. caused by severe wind gusts) results in a 231-degree phase shift of the received wave. This significantly distorts the received field in the antenna's focal plane, leading to SNR losses as severe as 4-6 dB under some conditions [7,44,47]. There is a need to recover as much signal energy as possible in order to realize the advantages of Ka-band communications.

A system developed at the Jet Propulsion Laboratory, the Array Feed Compensation System (AFCS), has been successful in SNR recovery, and the further development of algorithms and techniques for use with the AFCS to improve SNR performance lies at the heart of this research. The original development of the AFCS and its associated algorithms is first described in a paper by Vilnrotter and Rodemich [52] and expanded on later in a series of papers by Vilnrotter, Rodemich, Cramer, Iijima, Fort, Dolinar, Zohar, et al [6,7,20,38,46-48,48-51,53-55,57]. The AFCS, to be discussed further in Section 1.2, provides powerful real-time compensation for time-varying losses and pointing errors as well as compensation for many types of elevation-dependent static deformations [37,38,44]. Although parts of the static component can be compensated by the AFCS, there are parts of this component that must be compensated by means of a deformable mirror placed in the RF path whose deformations are determined using knowledge of systematic, gravity-induced deformations of the main reflector in order to insure that the received signal power can be captured and recombined by the AFCS [11,21,23,24,37,38,45]. The combination of both systematic error compensation with a deformable flat mirror and real-time and systematic error compensation with the AFCS yields tremendous performance enhancements [37,38,47].

In this dissertation, our emphasis is on the correction of errors in real-time using the AFCS. The DFP, used for compensating systematic errors as described in [38] and in the joint AFCS-DFP experiments conducted in 1999 [37,38,47], is highly efficient at recovering SNR that would otherwise be lost due to known, systematic deformations of the main reflector. Further information on the DFP and on joint AFCS/DFP experiments can be found in [11,21,23,24,35,37,38,44,45]. In this dissertation, the DFP is not considered, and our focus is on the use of the AFCS to compensate for real-time errors. This system is discussed in Section 1.2.

1.1 Overview of the antenna

NASA's Deep Space Network, hereafter referred to as the DSN, is a network of reflector antennas placed at three deep space communication facilities approximately 120^0 apart around the world; at Goldstone, California; near Madrid, Spain; and near Canberra, Australia. This strategic placement permits observation as the Earth rotates and enables the DSN to support interplanetary spacecraft missions, radio and radar astronomy observations, and some Earth-orbiting missions. The three 70-meter reflector antennas, one at each facility, are among the world's largest steerable reflector antennas and are the subject of this dissertation. Figure 1.1 is a simplified drawing of a DSN antenna which illustrates the coordinate system used throughout this work. Here, θ_{AZ} refers to the antenna's pointing azimuth; θ_{EL} refers to the pointing elevation; and the "focus" coordinate *Z* refers to the position of the antenna's subreflector along the pointing axis. Since this work deals with fine pointing, measured in millidegrees, as opposed to coarse pointing, the *(XEL,EL)*-plane is shown. This plane is used to specify very fine pointing in azimuth (*XEL*) and elevation (*EL*), and the planar approximation is accurate for the small pointing offsets under consideration.

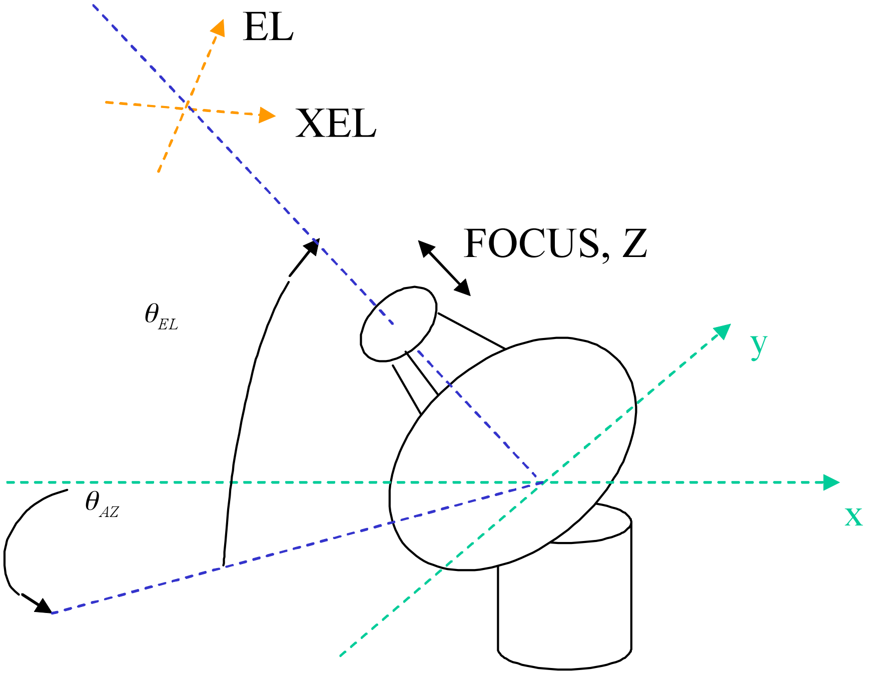

Figure 1.1: A simplified drawing of a DSN reflector antenna illustrating the coordinate system used in this research.

The main "dish" of the antenna is called the main reflector, while the secondary reflector above the center of the main dish is called the *subreflector*. An incoming RF wave first strikes the main reflector, which reflects the RF wave toward the subreflector. The subreflector, in turn, reflects the waves toward the focal plane of the antenna. Within the focal plane are one or more receiving horns which are designed to capture the RF energy and send it to the appropriate electronics for processing. This system is sensitive to changes in the shape of the main reflector and to changes in the position of the subreflector, both of which can have a significant effect on the antenna's overall gain performance and on the channel link budget [20,22-24,38-40,42,43,45-48,48-55]. In addition, errors in the antenna's pointing prediction model can reduce received SNR[28-31,47,57]. Our objective is to determine the pointing direction and the subreflector offset which will yield the highest possible received SNR and, thus, the highest possible antenna gain in the link budget. This problem therefore involves determining the current antenna offset *(XEL,EL,Z)* with respect to the optimal pointing and subreflector position, which is denoted as *(0,0,0)*. With this offset available, both the subreflector position and the pointing can be corrected to achieve maximum SNR in the focal plane.

1.2 Overview of the AFCS

The AFCS and its associated algorithms, developed at the Jet Propulsion Laboratory by Vilnrotter and Rodemich [52] and extended in [6,7,20,38,46-48,48-51,53-55,57], is a seven-element array of Ka-band horns placed in the antenna's focal plane. The primary effects of pointing errors and main reflector deformation are those of causing power to shift away from the nominal center of the focal plane and of distorting the focal plane field by spreading its power and introducing phase distortion[7,52]. An array of horns is better able to recover signal power that would otherwise be lost due to pointing errors and deformations of the antenna's main reflector than a single, large horn designed to capture the "average field" since there will be phase cancellations resulting in a loss of received SNR if a single large horn is used to capture such an "average field": hence, there is a need to use complex combining weights along with an array of horns for improved performance[7,52]. A photograph of the hexagonal array of AFCS receiving horns is shown in Figure 1.2. Here, the AFCS and its seven Ka-band horns are shown on the XKR cone of the 70-meter antenna at DSS-14 at Goldstone, California in the Mojave Desert. The seven Ka-band receiving horns are placed in the focal plane of the antenna.

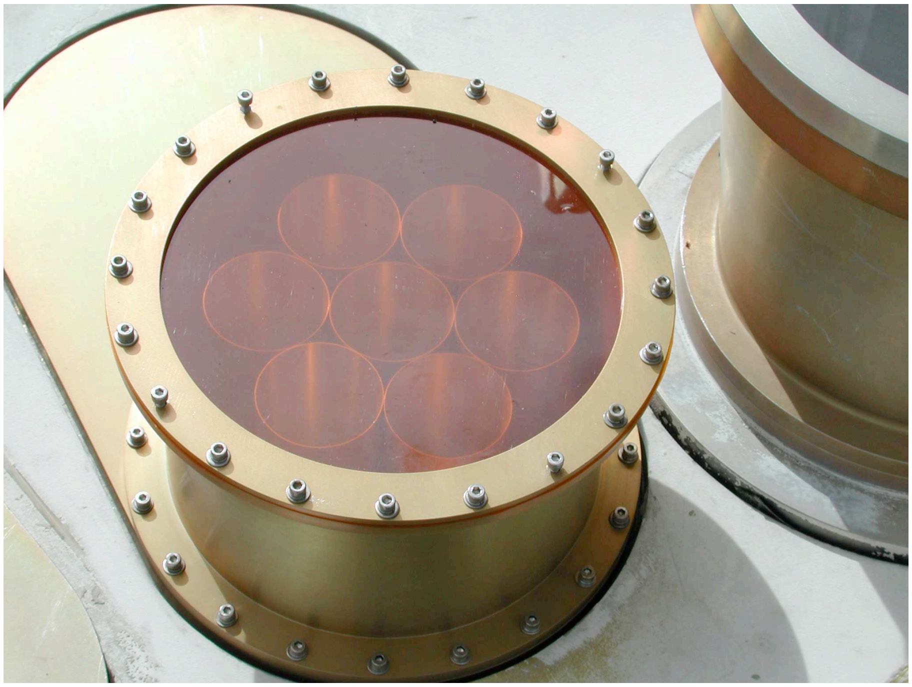

Figure 1.2: The Array-Feed Compensation System (AFCS).

Deformations of the dish have significant effects on the focal plane field. Figure 1.3 illustrates the focal plane field for the case of a perfect, undistorted dish with no pointing errors. Figure 1.4, Figure 1.5, and Figure 1.6 illustrate how the horns of the AFCS can recapture significant amounts of signal power that would otherwise be lost.

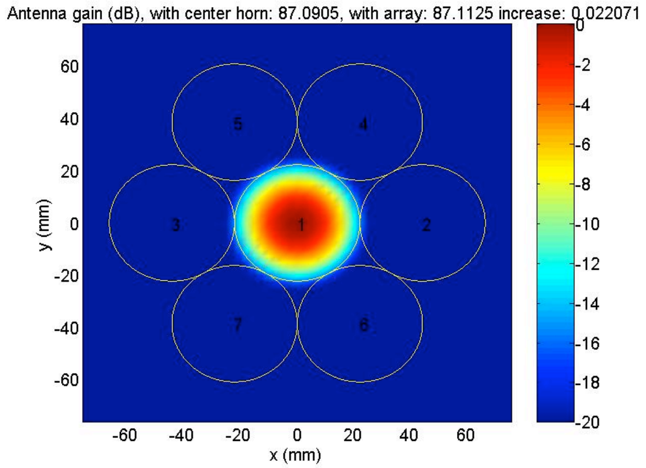

Figure 1.3: Focal plane field with no main reflector deformation.

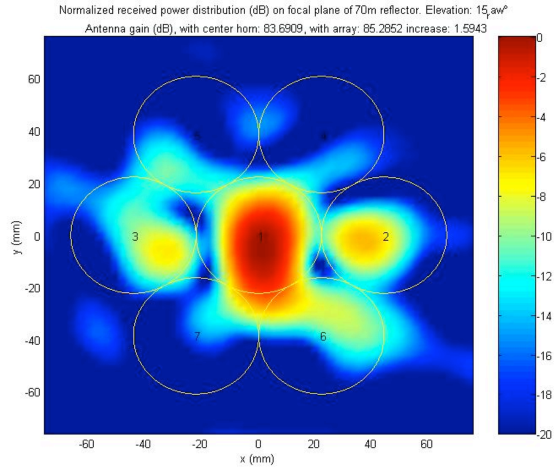

Figure 1.4: Power distribution in the focal plane at 15-degrees pointing elevation.

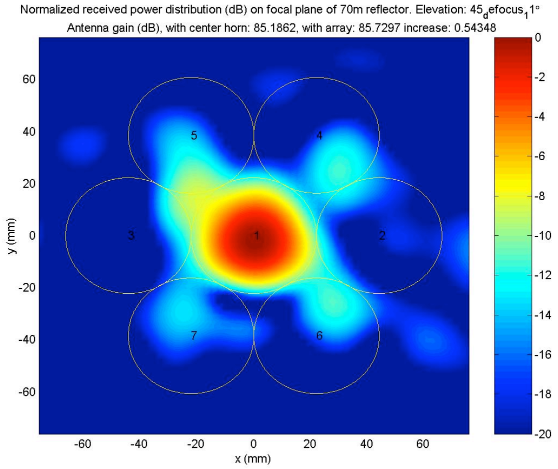

Figure 1.5: Power distribution in the focal plane field at 45-degrees pointing elevation.

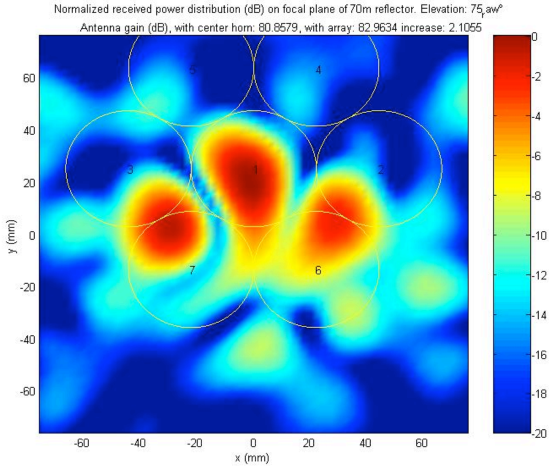

Figure 1.6: Power distribution in the focal plane field at 75-degrees pointing elevation.

The previous works by Vilnrotter and others [6,7,20,38,46-55] discuss the ability of the AFCS to recover the signal power captured by its outer horns and to use complex combining to create a single, combined channel which has a higher SNR than any one of the seven individual channels. Leading into the present work, Zohar and Vilnrotter [57] have studied the concept of using the complex baseband outputs of the AFCS to obtain information on the focal plane field which can be used to correct pointing, and Vilnrotter and Fort [47] have demonstrated a non-interpolated least-squares pointing correction algorithm. The present research focuses on the use of the AFCS to determine optimal pointing and optimal antenna subreflector position to maximize received SNR. The author's previous work on pointing correction, co-authored with Vilnrotter, Arabshahi, and Jamnejad, is in [28-31].

1.3 Overview of Dissertation Topics

This research concerns the following optimization problem: given the ability to observe the seven-channel output of the AFCS, how can we adjust both pointing and subreflector position to achieve the highest possible SNR? The inputs to this system are the pointing offsets *XEL* and *EL* and the subreflector *z*-axis position *Z*. The output is the SNR to be maximized.

In Chapter 2 we present the background material necessary for understanding both the problem itself and our approach to the problem. In Chapter 3, we define the optimization problem and consider some of the challenges involved in achieving maximum SNR with the AFCS and with the ability to control pointing and subreflector position. In Chapter 4, we present a constrained least-squares approach to controlling the pointing and subreflector position of the antenna, which leads to the interpolated least squares algorithms used in simulations and in real-world experiments. Chapter 4 differs from previous approaches to antenna pointing. Zohar and Vilnrotter had previously attempted to estimate pointing errors by estimating phase tilts in the aperture plane, and these tilts were computed on the basis of an inverse Fourier transform with aliasing taken into account [57]. Their approach yields important information on the nature of the problem, but it turns out to be impractical for the 70-meter antenna with its severe main reflector deformation. Others have studied the use of a monopulse system in controlling antenna pointing[12,15,27], but Zohar and Vilnrotter use a maximum likelihood approach that the monopulse does not yield [57]. A non-interpolated least-squares algorithm was used by Vilnrotter et al. in experiments conducted during 1999 [47], and the additional use of interpolation forms the backbone of Chapter 4. Chapter 5 discusses the use of neural networks on DSN antennas. This differs from previous approaches to the combined pointing and subreflector control problem on large DSN antennas in that it offers lower computational complexity than the interpolated least-squares algorithm of Chapter 4 and the potential for real-time adaptivity. Chapter 6 presents simulation results in which the interpolated least-squares algorithm and neural networks are compared and the advantages of neural networks are presented. Chapter 7 presents real-world experimental results taken from the 70-meter antenna at DSS-14 in 2001 and from the 34-meter antenna at DSS-13 in 2002. Chapter 8 presents our conclusions, along with several directions for future research.

2 Background material

2.1 DSN antenna overview

In our treatment of the antenna, we will make use of the theory of Fourier optics. Although Fourier optics and optical theory in general are typically applied to visible light systems, this theory remains applicable in many other regions of the electromagnetic spectrum. In particular, both geometric optics and wave optics can be applied to the study of DSN antennas since electromagnetic waves at RF frequencies, like visible light, are governed by the same laws of electromagnetism and propagation although these RF wavelengths are approximately four orders of magnitude longer than typical visible light wavelengths. Nevertheless, the antenna can still be viewed as an optical system that operates at RF frequencies rather than visible light frequencies, and we will refer to it as an *RF optical system.* The RF optics approach is widely used in the antenna engineering literature, and a very small sample of references that take this approach includes [8,11,19,21-24,26,35-44,44,45,57]. Many of these references are directly related to DSN antennas (although the textbook by Hecht [19] is meant to be generally applicable), and the RF optics approach is a widely accepted engineering approach.

The antenna under consideration is a reflector antenna whose RF optical system consists of a 70-meter diameter main reflector, a subreflector, and a focal plane in which receiving elements are placed. When a uniform plane wave from a source strikes the main reflector, it is focused onto the subreflector. The subreflector reflects the electromagnetic energy down to the focal plane of the antenna. Here, the energy collected by the 70-meter aperture is focused onto an area which is typically much less than one-hundred square centimeters. Receiving horns placed in the focal plane act as electromagnetic sensors which can capture the signal and send it down to high-frequency electronics for processing. A diagram of the overall system from [31] is shown in Figure 2.1.

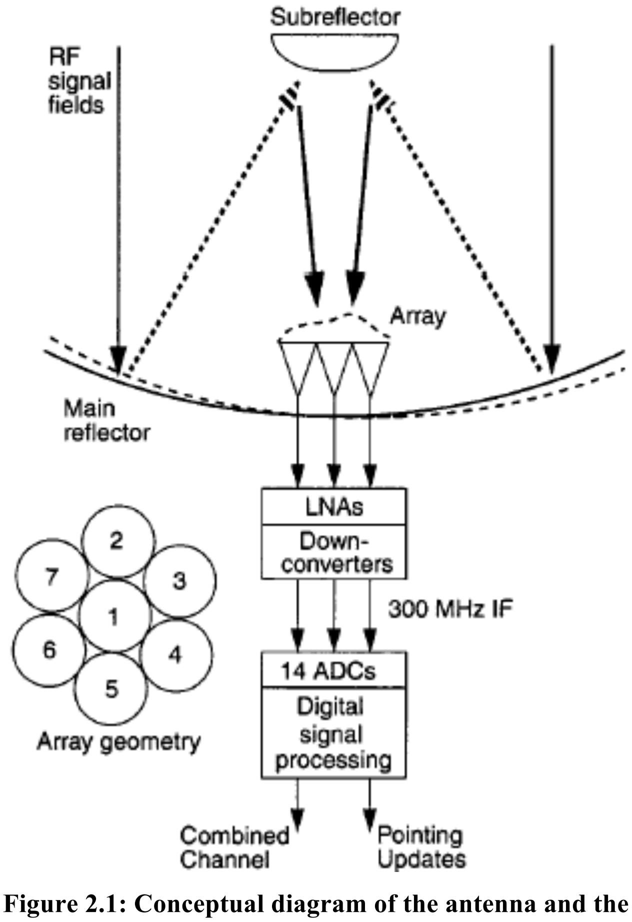

Figure 2.1: Conceptual diagram of the antenna and the AFCS. © 2002 IEEE. Reprinted with permission from Mukai, R., Vilnrotter, V.A., Arabshahi, P., and Jamnejad, V., "Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas", ***IEEE Transactions on Neural Networks*****, vol. 13, no. 5, pp. 1149-1162, Sep, 2002**

2.1.1 Important Relationships in a Cassegrain Antenna

For the purpose of this dissertation, it is important to document the model of the antenna that is used throughout much of the analysis. The modeling assumptions made here will play a role in understanding the relationship between subreflector position *Z* and equivalent deformation of the main reflector. These assumptions also play a role in understanding the relationship between dish deformations and resulting equivalent phase errors in the aperture plane and also enable a critique of the accuracy of our model. The model developed here will play a role in determining the form of an interpolated least squares estimator to be developed in Chapter 4.

Cassegrain antenna systems, including the 70-meter DSN antenna under study, are sometimes modeled as paraboloid-hyperboloid reflector systems, with the main reflector modeled as an ideal, circularly symmetric paraboloid and the subreflector modeled as an ideal, circularly symmetric hyperboloid [19,32,39], leading to a Fourier-transform model[57]. Although the actual 70-meter DSN antenna is a shaped reflector system and not a true paraboloid-hyperboloid Cassegrain [2,23,39], it is possible to treat the system as if it were based upon a parabolic main reflector and a hyperbolic subreflector, and this approach is taken here. This is an approximation to the true antenna. Simulations that compare the paraboloid-hyperboloid model to a physically realistic model with respect to the problem of pointing and focal correction are part of the subject of Chapter 6. In this chapter and in Chapters 3 and 4, the paraboloid-hyperboloid model will be implicitly assumed as it sometimes is in the literature. For example, this model is used by Zohar and

Vilnrotter in their analysis of pointing [57], and a modified form of this model, with corrections to take the true reflector shapes into account, is used in holography by Rochblatt et al. [39,40,42,43].

A paraboloid-hyperboloid Cassegrain system yields optical behavior equivalent to that of a pure parabolic system with the same aperture but with a longer parabolic focal length [25]. Define:

1. f : The *effective focal length* of the equivalent parabolic reflector
2. f_p : The actual focal length of the parabolic reflector in the Cassegrain system

The *magnification* of the system is defined by:

$$M = \frac{f}{f_p} \tag{0.0.1}$$

Parabolic deformation of the main reflector can be modeled by a change in the focal length of the parabola. Let *z* be the axial coordinate and let *r* be the radial coordinate. To model this change, start with the equation for an ideal parabolic reflector:

$$r^2 = 4f\,z \tag{0.0.2}$$

where *f* is the focal length of the circularly symmetric paraboloid. Equation (0.0.2) can be used to compute the shape of either the actual reflector or the effective (equivalent) parabolic reflector, depending on whether we use the true focal length or the effective focal length. The parabolic reflector shape is shown in Figure 2.2.

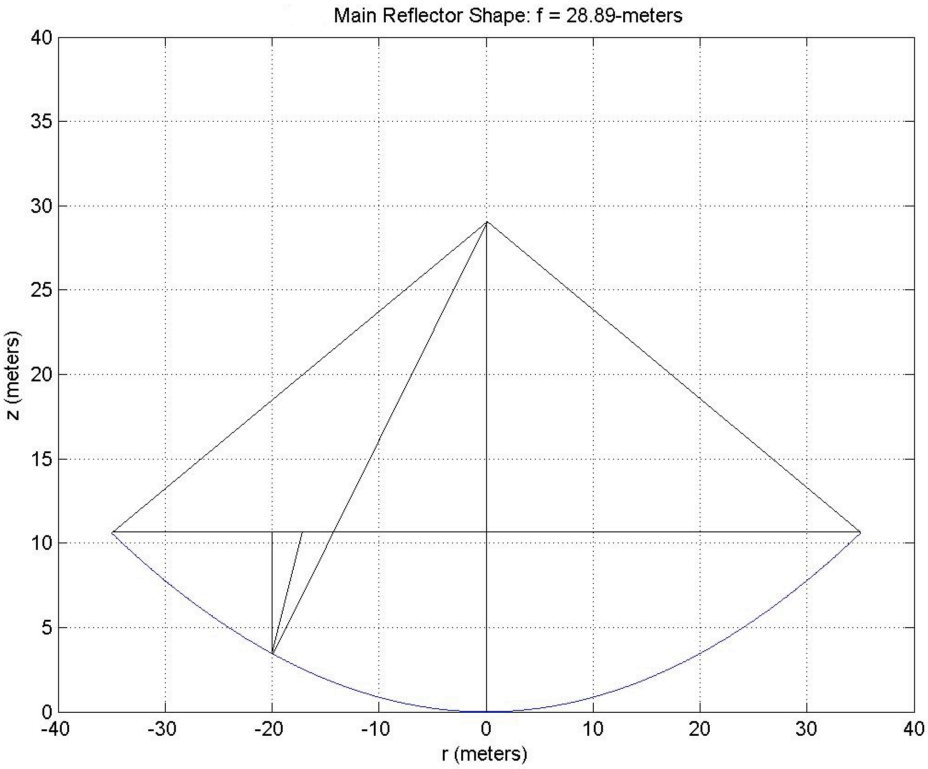

Figure 2.2: Parabolic reflector shape

The phase errors that result from deformation of the main reflector were analyzed extensively by Rochblatt et al. for the purpose of determining antenna deformations via holography [37,39-44], and a result from that work states that phase errors (ψ) caused by deformation of the main reflector are given by [43]:

$$\psi = \frac{4\pi}{\lambda}\varepsilon\cos\phi \qquad (0.0.3)$$

where ε is the error normal to the antenna surface and where the angle ϕ is as shown in Figure 2.2. Consider the value of ϕ at the edge of the main reflector, which yields the greatest value of $\cos\phi$. For the true main reflector, we have:

$$\phi = \frac{1}{2}\arctan\left(\frac{35}{28.89 - \frac{35^2}{4\cdot 28.89}}\right) \approx 31.21° \qquad (0.0.4)$$

which yields $\cos\phi = 0.8553$. For the effective main reflector, we have:

$$\phi = \frac{1}{2}\arctan\left(\frac{35}{152 - \frac{35^2}{4\bullet 152}}\right) \approx 6.568° \qquad (0.0.5)$$

which yields $\cos\phi = 0.9934$.

For a given error ε measured normal to the surface, the change in overall RF path length is only 13.9% less for the true main reflector than it is for the effective reflector. However, in this dissertation, all errors are measured along the *z*-direction instead of normal to the surface, which means [43]:

$$\Delta z = \varepsilon \cos\phi \qquad (0.0.6)$$

Since the phase error given in equation (0.0.3) is linearly proportional to the *z*-direction error given by equation (0.0.6), and since we only deal with Δz directly without concerning ourselves with errors normal to the surface [39-43], we can treat errors in main reflector shape along the *z*-direction as being the same whether we are examining the true main reflector or the effective main reflector. An error in *z*-position always has the same effect on phase. Since there is no need to concern ourselves with whether a deformation applies to the true main reflector or to the effective main reflector that yields the same focal plane field as the paraboloid-hyperboloid Cassegrain system, we will treat the effective main reflector throughout this dissertation with the understanding that *small* z-deformations of the true main reflector have the same phase effects as if they had been applied to the effective main reflector.

2.2 Predict models

Predict models are sets of data supplied by the DSN that are used to predict carrier frequencies seen on the ground, proper antenna pointing, and proper subreflector positioning. Such predict models are described by Vilnrotter and Fort in [47] and play an important role in the operation of the antenna.

The received frequency at the antenna is affected by the relative movement of the spacecraft with respect to the ground receiving station. The Earth's rotation on its axis and revolution around the sun along with motion of the spacecraft itself lead to time-variations in the carrier frequency received at an Earth-based ground station due to the Doppler effect [47], and the DSN computes the received frequency at each ground station as a function of time. This set of frequency predicts is referred to as a set of *sky frequency predicts*, and knowledge of these predicts plays an important role in enabling ground receivers to acquire the carrier and lock on to it. As stated by Vilnrotter and Fort [47], the AFCS receiver uses these predicts in order to acquire and tracking the incoming carrier signal from the spacecraft, which was Cassini in the case of the real-world experiments presented in Chapter 7 and in [31].

Two other predict models, one for pointing and another for the subreflector, play an important role in practical operations involving the AFCS. Given Earth's rotation and the trajectory of the spacecraft, it is possible to compute a set of *pointing* *predicts* which permit the antenna to track the spacecraft with errors no greater than a few millidegrees (thousandths of a degree). These highly accurate pointing models provide the primary source of pointing information for the antenna. Since Earth's rotation typically requires the antenna to move at an average rate of about four millidegrees per second during tracking, primary pointing information must come from the predicts. However, since pointing predicts typically have errors of a few millidegrees, it is necessary to refine the antenna's pointing in order to achieve maximum signal power at the focal plane [47,57]. This, in turn, requires that small pointing corrections in the *(XEL, EL)* plane shown in Figure 1.1 be applied to maintain maximum SNR at the receiver, and our discussion of antenna pointing focuses on this problem. A good pointing predict model which requires only small corrections of a few millidegrees is assumed throughout this dissertation.

A DSN-provided *subreflector model*, which is used to position the subreflector in as close to an optimal position as possible during tracking, is used to control the positioning of the subreflector. We focus only on the subreflector's position along the *Z*-axis shown in Figure 1.1. The subreflector model is typically accurate, but we seek to find corrections to subreflector position since errors in position along the *Z*-axis cause errors in focus at the focal plane. Such errors tend to disperse energy over a larger area of the focal plane, causing SNR losses [43]. Additionally, focal errors may be caused by wind, thermal expansion, and by the very weight of the dish, especially at high pointing elevations (high values of θ_{EL}) [47], and a 5-millimeter error in subreflector position can cause a loss of roughly 1.4 dB

[22,39-44]. For this reason, the ability to estimate the optimal subreflector position to maximize received signal power is also important [43]. Systematic subreflector error correction is considered by Rochblatt et al. [39,40,42-44], but there is also a need for real-time estimation of optimal subreflector position that is addressed here.

2.3 AFCS Signal Modeling

AFCS signal modeling is described in the original AFCS paper [52] and also in [46,50,51,53,54,57]. A detailed description of AFCS signal processing is also given by Vilnrotter and Fort in [47], and we present a summary of the information on AFCS signal processing from [47] here. The outputs of the AFCS are used in determining and in compensating for both pointing errors and certain dish deformations. Each of the seven horns of the AFCS receives a portion of the signal. The signal in the *kth* channel is written [31,46,47,52-54]:

$$r_k(t) = s_k(t) + n_k(t) \quad k = 1,2,3,\ldots,7 \tag{0.0.7}$$

where the signal and background noise components (real signals) are given by [31,46,47,50-54]:

$$s_k(t) = S_k \cos(\omega t + \theta_k) \tag{0.0.8}$$

and

$$n_k(t) = \left[n_{ck}(t)\cos(\omega t) - n_{sk}(t)\sin(\omega t)\right] \tag{0.0.9}$$

Here, $n_{ck}(t)$ and $n_{sk}(t)$ are uncorrelated baseband random processes which represent the in-phase and quadrature components of the noise, and ω represents the 32-GHz carrier.

As explained in [47], a 31.7-GHz signal is used to downconvert the 32-GHz signal to an intermediate frequency of 300-MHz. From this point, since the received carrier frequency is affected by both Earth's rotation and the relative motion of the spacecraft, a set of pre-calculated spacecraft *frequency predicts* is used to determine the proper mixer frequency, which is also tracked by a frequency-locked loop. The final downconversion yields a set of complex baseband samples. The system samples at a rate of 128 samples per second, and these samples are represented as:

$$\tilde{r}_k(i) = \tilde{s}_k(i) + \tilde{n}_k(i) \quad k = 1,2,3,\ldots,7 \quad i \in \mathbb{N} \tag{0.0.10}$$

where

$$\tilde{s}_k(i) \triangleq S_k e^{j\theta_k} \tag{0.0.11}$$

The noise statistics are given by:

$$E\left[\tilde{n}_k(i)\right] = 0 \tag{0.0.12}$$

and

$$E\left[\left|\tilde{n}_k\right|^2\right] = 2\sigma^2 \tag{0.0.13}$$

It is assumed that noise is independent among channels and that successive noise samples in any given channel are independent. The amplitude and phase of the signal in equation (0.0.11) are assumed constant from sample to sample. A received vector is defined by:

$$\tilde{\mathbf{r}}(i) = \left[\tilde{r}_1(i) \quad \tilde{r}_2(i) \quad \ldots \quad \tilde{r}_7(i)\right]^T \tag{0.0.14}$$

The signal represented by the vector in equation (0.0.14) can be separated into real and imaginary parts as follows. Let:

$$\begin{aligned} v_{2k-1}(i) &= \mathrm{Re}\left[\tilde{r}_k(i)\right] \\ v_{2k}(i) &= \mathrm{Im}\left[\tilde{r}_k(i)\right] \end{aligned} \qquad (0.0.15)$$

Then define:

$$\mathbf{v}(i) = \left[v_1(i) \quad v_2(i) \quad \ldots \quad v_{14}(i)\right]^T \qquad (0.0.16)$$

The variance of each of the components of vector $\mathbf{v}(i)$ is σ^2. This vector is corrupted by zero-mean AWGN (additive white Gaussian noise). At the operating point *(XEL,EL,Z)* that specifies the antenna's pointing and subreflector position, the mean of $\mathbf{v}(i)$ is the fourteen-element vector

$\vec{\mu}(XEL, EL, Z) = \left[\mu_1(XEL, EL, Z) \quad \mu_2(XEL, EL, Z) \quad \ldots \quad \mu_{14}(XEL, EL, Z)\right]^T$ and the probability density function is:

$$\begin{aligned} f(\mathbf{v}(i)) = {} & \frac{1}{\sqrt{(2\pi)^N \det(\Lambda)}} \\ & \exp\left(\begin{aligned} & -\frac{1}{2}\left(\mathbf{v}(i) - \vec{\mu}(XEL, EL, Z)\right)^T \Lambda^{-1} \\ & \left(\mathbf{v}(i) - \vec{\mu}(XEL, EL, Z)\right) \end{aligned} \right) \end{aligned} \qquad (0.0.17)$$

where *N* is the length of the vector (in this case, *N=14*) and where

$$\mathbf{\Lambda} = \begin{bmatrix} \sigma^2 & 0 & \ldots & 0 \\ 0 & \sigma^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma^2 \end{bmatrix} \qquad (0.0.18)$$

In order to reduce the effects of noise, averaging may be used as follows:

$$\begin{aligned}\mathbf{v}_a(jL) &= \left[v_{a,1}(jL) \quad v_{a,2}(jL) \quad \cdots \quad v_{a,14}(jL) \right] \\ &= \frac{1}{L} \sum_{i=jL-L+1}^{jL} \mathbf{v}(i)\end{aligned} \qquad (0.0.19)$$

A new vector $\mathbf{v}(i)$ is generated 128 times per second [47]. By averaging with a period $L = 128$ we can generate a new vector once per second. An averaging period of $L = 1280$ results in a new vector every ten seconds, and so on. The effects of noise are reduced since:

$$\operatorname{var}\left[v_{a,k}(jL) \right] = \frac{\sigma^2}{L} \qquad (0.0.20)$$

As shown by Zohar and Vilnrotter [57] and as will be discussed in Chapters 3 and 4, this vector of samples from the AFCS yields important information on the focal plane field. It is this information which can be used to correct pointing and focal errors in order to maximize the received SNR.

For notational convenience we will sometimes denote the average AFCS complex baseband output vector $\mathbf{v}_a(jL)$ simply as $\mathbf{v}$, the complex baseband output vector. This will be true especially in our discussion of interpolated least-squares estimation in Chapter 4.

2.4 Neural networks

Neural networks provide a way to approximate continuous functions over bounded intervals and are a significant part of this research. Two types of neural networks, radial basis function networks and multilayer perceptrons, were used. Their properties are discussed in the next two subsections.

If we are seeking to approximate a continuous scalar function $y = g(\mathbf{x})$, we can often train a neural network to do so if we have a large number of training samples. Let $\{\mathbf{x}_n\}$ be a set of *N* training input vectors and $\{d_n\}$ be the corresponding set of desired outputs where $d_n = g(\mathbf{x}_n)$. Define y_n to be the neural network's response to input vector $\mathbf{x}_n$. The mean squared error over the training set is:

$$E = \frac{1}{N}\sum_{n=1}^{N}\left|d_n - y_n\right|^2 \tag{0.0.21}$$

The objective of neural network design and training is that of minimizing the mean squared error defined in equation (0.0.21). Minimization of this mean squared error results in a network that provides a better approximation of the underlying function $y = g(\mathbf{x})$, and this is the subject of the next two subsections.

2.4.1 Radial Basis Function networks

Radial basis function (RBF) networks are one of two types of neural networks evaluated in both simulations and real-world experiments for antenna pointing control. These networks are described in detail by Haykin [17], and the OLS learning algorithm used to train them in this research was developed by Chen et al. [5]. Here, we will summarize the material presented by Haykin [17]. A diagram of such a network is shown in Figure 2.3.

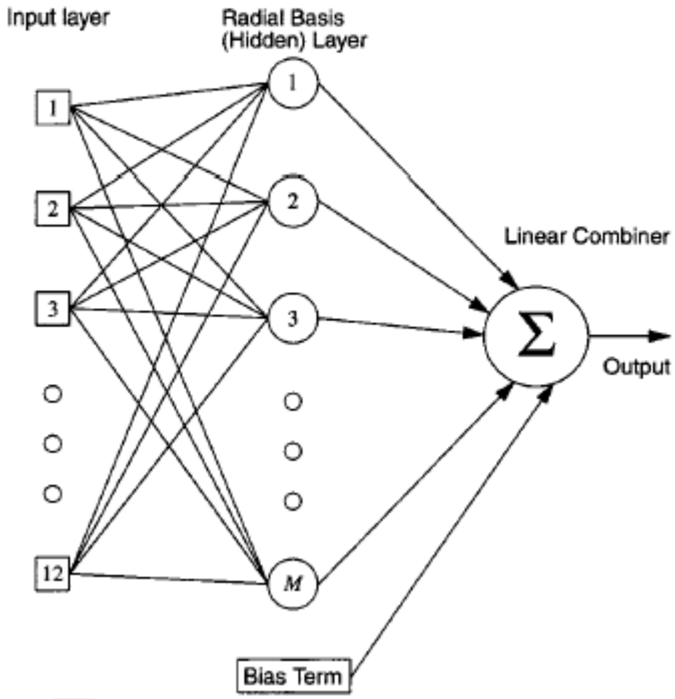

Figure 2.3: Diagram of a single output RBF neural network. © 2002 IEEE. Reprinted with permission from Mukai, R., Vilnrotter, V.A., Arabshahi, P., and Jamnejad, V., "Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas", *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1149-1162, Sep, 2002

In Figure 2.3, there are twelve scalar inputs to the RBF network. Note that an RBF network may have an arbitrary number of inputs in general, although twelve-input networks were used in this research for reasons to be discussed in Chapter 4. Let these be denoted by $x_1, x_2, \ldots, x_{12}$. We note that each of the *M* radial basis units receives all twelve of these inputs. The scalar inputs may be combined into a vector:

$$\vec{\mathbf{x}} = \begin{pmatrix} x_1 & x_2 & \dots & x_{12} \end{pmatrix} \tag{0.0.22}$$

Each radial basis unit receives the complete input vector $\vec{\mathbf{x}}$, and the $n^{\underline{th}}$ basis unit has its own *center vector* defined by:

$$\vec{\mu}_n = \begin{pmatrix} \mu_{n,1} & \mu_{n,2} & \dots & \mu_{n,12} \end{pmatrix} \tag{0.0.23}$$

which is of the same dimensionality as the input vector $\vec{\mathbf{x}}$. The $n^{\underline{th}}$ basis unit performs the following computation:

$$z_n = \exp\left(-\left(b\left\|\vec{\mathbf{x}} - \vec{\mu}_n\right\|\right)^2\right) \tag{0.0.24}$$

where b is a parameter that determines the region of response of the basis unit. If $\vec{\mathbf{x}} = \vec{\mu}_n$, the unit's response is unity. As the Euclidean distance between $\vec{\mathbf{x}}$ and $\vec{\mu}$ approaches infinity, the unit's response approaches zero. One way to define the region of response is to define the Euclidean distance between $\vec{\mathbf{x}}$ and $\vec{\mu}$ such that the unit's output is 0.5. This distance, which is sometimes called the *spread*, is:

$$\sigma_{spread} \triangleq \frac{\sqrt{\ln(2)}}{b} \tag{0.0.25}$$

since

$$\left\|\vec{\mathbf{x}} - \vec{\mu}_n\right\| = \sigma_{spread} \Rightarrow \exp\left(-\left(b\left\|\vec{\mathbf{x}} - \vec{\mu}_n\right\|\right)^2\right) = \frac{1}{2} \tag{0.0.26}$$

The outputs of the radial basis units are sent to the linear combiner shown in Figure 2.3. The output of a single-output network is:

$$y = bias + \sum_{n=1}^{M} w_n z_n \tag{0.0.27}$$

Although Figure 2.3 shows only a single output of the form given by equation (0.0.27), some RBF networks have multiple outputs, each with its own independent bias term and set of linear combining weights.

RBF networks can be used in classification problems, and Cover's theorem on the separability of patterns shows us that as long as a set of vectors is separable, meaning that there exists a hypersurface capable of separating them into a given desired binary partition, there will always exist an RBF network capable of performing this desired partitioning in the sense that its output will be greater than or equal to zero for one class of inputs and less than zero for the other class of inputs [17]. RBF networks are also very powerful function interpolators. Given a set of data points, we may wish to approximate the underlying function responsible for generating those points. The theory of regularization, which is not discussed further in this dissertation, gives us tools for creating RBF networks capable of very accurate function approximation and interpolation given a set of data points [17], and it is this function approximation capability that is exploited here.

Constructing an RBF network to perform the desired interpolation is a challenging problem, and one solution that often works well in practice is the OLS learning procedure developed by Chen, et al. [5]. A description of this procedure, based on the original paper by Chen, et al. [5], is given in Appendix A.

OLS has a number of useful properties. It is a computationally fast network design algorithm which does not have the severe computational overhead of gradient backpropagation methods [5,17]. If the input vectors are indeed statistically representative of the underlying input process, it often does an excellent job of finding a set of radial basis centers such that the number of radial basis units is minimized even while the mean squared error criterion is met [5]. The networks produced have, in both simulation and real-life, yielded very good antenna pointing control performance as will be seen in Chapters 6 and 7 and as shown in [28-31].

2.4.2 Multilayer feedforward networks

In our treatment of multilayer feedforward networks, we will closely follow Hagan, Demuth, and Beale [16], whose highly efficient matrix notation both simplifies the treatment of these networks and is used here. A single hidden layer feed forward network is similar to the RBF network of Figure 2.3 but with the radial basis units replaced by a linear combiner with an output non-linearity. The *n<u>th</u>* hidden layer can be described by [16]:

$$\mathbf{a}_n = \mathbf{f}_n\left(\mathbf{W}_n \mathbf{a}_{n-1} + \mathbf{b}_n\right) \qquad (0.0.28)$$

where:

1. $\mathbf{a}_n$ is the column vector of outputs of the *n<u>th</u>* layer for $n \geq 1$
2. $\mathbf{a}_0$ is the column vector of inputs to the network
3. $\mathbf{W}_n$ is the synaptic weight matrix of the *n<u>th</u>* layer
4. $\mathbf{b}_n$ is the column of bias weights of the *n<u>th</u>* layer
5. $\mathbf{f}_n$ is a function applied to the column vector $\mathbf{W}_n \mathbf{a}_{n-1} + \mathbf{b}_n$ on an element-by-element basis. It may be linear only in the output layer and will be non-linear for all hidden layers. A common choice is the hyperbolic tangent function [16,17], and the hyperbolic tangent was used in the hidden layers of all multilayer feedforward networks discussed in this dissertation.

It should be noted having a linear hidden layer *n* is not sensible. To see this, let:

$$\mathbf{a}_n = \mathbf{W}_n \mathbf{a}_{n-1} + \mathbf{b}_n \tag{0.0.29}$$

thus making layer *n* linear. In that case, we can compute the activation levels of layer *(n+1)* as follows:

$$\begin{aligned}\mathbf{a}_{n+1} &= \mathbf{f}_{n+1}\left(\mathbf{W}_{n+1}\mathbf{a}_n + \mathbf{b}_{n+1}\right) \\ &= \mathbf{f}_{n+1}\left(\mathbf{W}_{n+1}\left(\mathbf{W}_n \mathbf{a}_{n-1} + \mathbf{b}_n\right) + \mathbf{b}_{n+1}\right) \\ &= \mathbf{f}_{n+1}\left(\mathbf{W}_{n+1}\mathbf{W}_n \mathbf{a}_{n-1} + \mathbf{W}_{n+1}\mathbf{b}_n + \mathbf{b}_{n+1}\right)\end{aligned} \tag{0.0.30}$$

It is now clear that we can replace layers *n* and *(n+1)* with a single hidden layer with $\mathbf{W}_{n,new} = \mathbf{W}_{n+1}\mathbf{W}_n$ and with $\mathbf{b}_{n,new} = \mathbf{W}_{n+1}\mathbf{b}_n + \mathbf{b}_{n+1}$. Thus, a linear hidden layer could easily be incorporated into the succeeding layer. This would permit the elimination of any linear hidden layers, and a purely linear network would always have an equivalent single-layer form. In this treatment, it will be assumed that all hidden layers have a non-linear function $\mathbf{f}_n$.

Multilayer feedforward networks are commonly used for function approximation. A very important theorem regarding these networks is the Universal Approximation Theorem [9,10,16,17]. Consider a function $\mathbf{y} = \mathbf{f}\left(\mathbf{x}\right)$ where both the input and the output may, in general, be vectors. Assume that each element of this function's output is a continuous, bounded function over a finite, bounded region of the input vector space. For any finite, positive error tolerance ε there exists a multilayer feedforward network with a single, non-linear hidden layer such that $\left\|\mathbf{f}\left(\mathbf{x}\right) - \mathbf{y}_{net}\right\| < \varepsilon$ for all input from the bounded region where $\mathbf{y}_{net}$ is the output of the network corresponding to $\mathbf{x}$ [9,10,16,17].

The Universal Approximation Theorem illustrates the function approximation power of single-hidden-layer networks, but it is only an existence theorem. It does not give us a procedure for actually finding such a network [9,10,16,17]. In particular, it does not tell us the number of hidden layer units needed, the necessary synaptic weight matrix, or the necessary bias matrix. In most cases, the number of hidden layer units is determined a priori, sometimes on the basis of experiment and sometimes on the basis of formal criteria. The synaptic weights and the bias terms are usually determined through an iterative training process. The backpropagation method is the most commonly used form of training, and the basic version of backpropagation is described by both Hagan et al. [16] and Haykin [17] among many sources. We present a discussion of backpropagation that closely follows Hagan et al. [16] and uses the same notation in Appendix B.

2.5 Summary

This chapter introduces a number of concepts that are needed in subsequent chapters. The DSN antenna studied in this dissertation can be enhanced using a seven-element focal plane array system that uses complex baseband downconversion to obtain seven complex baseband signals, one for each focal plane element. These signals are processed in order to obtain useful information on antenna operating conditions. Although predict models provide sky frequency information, pointing information, and subreflector position information to be used in communications with distant spacecraft, predicts often contain errors that can lead to a loss of SNR. The complex baseband signals provided by the AFCS will be exploited to achieve more accurate pointing and subreflector control.

Neural networks provide a powerful method of approximating smooth, continuous, and bounded functions. Since neural networks are trainable, they also offer the advantage of adaptivity under changing conditions. Both neural networks and interpolated least squares estimation methods to be introduced in Chapter 4 will be evaluated in this dissertation.

3 Defining the Problem: Optimizing SNR with Pointing and Subreflector

Our objective is that of achieving maximum SNR at the receiver given the ability to control pointing and the *z*-axis position of the antenna's subreflector. We begin by modeling the antenna as a system whose inputs are the pointing and subreflector controls *(XEL, EL, Z)* and whose outputs are the complex baseband outputs of the AFCS. Since we assume the system noise to be given by the Gaussian distribution of equation (0.0.17), and since this Gaussian noise is independent of the received signal, the problem of maximizing the SNR is equivalent to that of maximizing received signal power. The optimization problem we wish to solve is this: find the pointing offsets and subreflector position, which are given in the ordered triple *(XEL, EL, Z)*, such that the power received by the AFCS in the focal plane is maximized. In other words, we are seeking to maximize a function *P(XEL, EL, Z)* which gives power as a function of three inputs.

One complication here is that the function *P(XEL,EL,Z)* is itself dependent on overall pointing elevation, which we distinguish from the fine-pointing coordinate *EL* by calling it θ_{EL}. From this point forward, *EL* will be used to denote changes in fine pointing relative to the spacecraft's position, while θ_{EL} will be used to denote overall pointing elevation of the antenna itself. We note that θ_{EL} has a significant effect on the received focal plane field as it results in significant changes in the deformation of the main reflector [24,37,39,40,42-44]. The resulting changes in the focal plane field are illustrated from Figure 1.4 through Figure 1.6 for 15, 45, and 75 degrees. This implies that the power function to be maximized changes with elevation. Our optimization problem, therefore, is that of maximizing $P_{\theta_{EL}}(XEL, EL, Z)$ for any given pointing elevation θ_{EL}.

It should be noted that $P_{\theta_{EL}}(XEL, EL, Z)$ is the combined power of all seven channels. As first discussed in [52] and also in [46,47,50,51,53,54] and in Section 3.5, this is a function not only of the offset *(XEL,EL,Z)* but also of a set of complex combining weights. The problem of estimating these weights and of obtaining maximum likelihood estimates of these weights has been solved in [46,47,51-54], and the effect of these complex combining weights on the overall optimization problem is discussed in Section 3.5.

In this chapter and in Chapter 4, we will be relying on the equivalent parabolic reflector model of the antenna discussed in Section 2.1.1.

3.1 The aperture plane field

In order to express $P_{\theta_{EL}}(XEL, EL, Z)$, we need to start with the relationship between the field at the aperture plane and the field seen by the AFCS at the focal plane. In the paraboloid-hyperboloid model, this is a Fourier transform relationship [57], and we will closely follow Goodman [14] along with Zohar and Vilnrotter [57] in our presentation of this relationship. For simplicity, polar coordinates will be used throughout much of our treatment. The coordinate pair $\left(r,\theta\right)$ will always refer to the aperture plane, where the aperture is that of the antenna's main 70-meter diameter reflector. The coordinate pair $\left(\rho,\phi\right)$ will always refer to the focal plane, which is the plane in which the AFCS horns are placed. Let $U_a\left(r,\theta\right)$ be the scalar field in the aperture plane, and let $U_f\left(\rho,\phi\right)$ be the scalar field in the focal plane. Goodman gives the following relationship between the two [14]:

$$U_f\left(\rho,\phi\right)=\frac{\exp\left(j\frac{k}{2f}\rho^2\right)}{j\lambda f}\int_0^R\int_{-\pi}^{\pi}U_a\left(r,\theta\right)\exp\left(-j\frac{2\pi}{\lambda f}r\rho\cos\left(\theta-\phi\right)\right)rd\theta dr \qquad (0.0.31)$$

where we use the following definitions:

1. $U_a\left(r,\theta\right)$: The scalar field in the aperture plane.
2. λ: The wavelength of the carrier. 9.369-mm here.
3. k: The *wave number*, defined by $k = {}^{2\pi}\!/_{\lambda}$.
4. f: The effective *focal length* of the antenna. 152-meters.
5. R: The radius of the main reflector. 35-meters.

It must be noted that the shaped-reflector antenna does not have a well-defined effective focal length [2]. The use of 152-meters as an effective focal length is based on a "best-fit" model. A physical optics simulation of the actual antenna was written by Dr. Vahraz Jamnejad of the Jet Propulsion Laboratory, and this simulation was used to generate the focal plane field for an ideal, undistorted 70-meter antenna. Focal lengths ranging from 120-meters to 200-meters in 1-meter increments were tested using equation (0.0.31). In each case, the resulting Fourier-transform-based focal plane field $U_f\left(\rho,\phi\right)$ was compared to the physical optics generated field, and a squared error distance measure between field magnitudes was used as an error metric. The field generated using f=152-meters was found to yield the closest match to the field generated by the physical optics simulation, and f=152-meters was taken to be the effective focal length of our idealized Fourier-transform based model. It must be noted that this figure differs from what one would obtain by looking at the angle, as seen from the focal plane, subtended by the antenna's subreflector, and this is in agreement with the fact that the actual shaped-reflector antenna (as opposed to an idealized paraboloid-hyperboloid model) lacks a well-defined effective focal length [2]. Since the Fourier-transform based model is an approximation to the true antenna, and since our theoretical development is based upon this model, comparison simulations were run in order to determine the applicability of this model to the optimization problem at hand. The results of these simulations are given in Chapter 6.

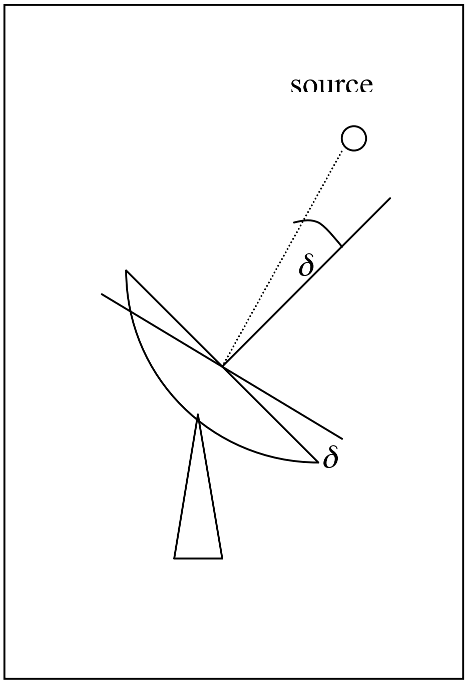

Figure 3.1: Illustration of the antenna with a pointing error.

It is assumed that the source, which was the Cassini spacecraft in the real-world experiments of Chapter 7 and of [31], illuminates the antenna with uniform plane wave illumination. Let δ_{XEL} and δ_{EL} denote the pointing errors, in radians, in the *XEL* and *EL* directions, respectively. Let r and θ be the polar coordinates of the aperture plane. The uniform plane wave, as received in the aperture plane of the antenna, is given by [57]:

$$U_a(r,\theta) = \exp\left(j\frac{2\pi}{\lambda} r\left(\delta_{XEL}\cos(\theta) + \delta_{EL}\sin(\theta)\right)\right) \qquad (0.0.32)$$

where the small angle approximation is used under the assumption that these are small pointing errors.

The field given in equation (0.0.32) is the field that impinges upon the aperture plane of the antenna in the presence of pointing errors. If the pointing errors are zero, then the scalar field is a constant over the entire aperture plane (at complex baseband) [14,57].

The deformation of the *effective* antenna reflector, given by the function $d(r,\theta)$ in the aperture plane coordinate system, describes deformations of the dish caused by wind, gravity, thermal gradients, and other imperfections in the shape of the main reflector [22,37,39,40,42-44], although the deformations in the aforementioned references were measured both normal to the surface and along the *z*-direction. This dissertation focuses only on *z*-direction errors. It is possible to model the effects of this deformation by having the distortion affect the aperture plane field instead of the main reflector. That is, we can assume a perfect main reflector with phase distortions in the aperture plane field proportional to $d(r,\theta)$. In doing this we note that the deformations occur internal to the optical system of the antenna. The phase error caused by any deformation of the main reflector is doubled by reflection [39,40,42,43]. Phase errors are given by $\frac{4\pi}{\lambda}d(r,\theta)$ instead of $\frac{2\pi}{\lambda}d(r,\theta)$ due to the reflection doubling effect. We can now write the *effective aperture plane field* in terms of both pointing errors and deformations as:

$$U_a(r,\theta)=\exp\left(j\left(\left(\frac{4\pi}{\lambda}d(r,\theta)\right)+\left(\frac{2\pi}{\lambda}r\left[\delta_{XEL}\cos(\theta)+\delta_{EL}\sin(\theta)\right]\right)\right)\right) \quad (0.0.33)$$

To simplify the treatment, we will combine the deformation and pointing terms into one function, placing both in $d(r,\theta)$. This does require multiplying the pointing error terms by $\frac{1}{2}$ prior to adding them in to the new $d(r,\theta)$ since pointing errors should not be phase doubled while dish deformations are phase doubled. The resulting equation for the aperture plane field, based on this effective $d(r,\theta)$, is:

$$U_a(r,\theta)=\exp\left(j\frac{4\pi}{\lambda}d(r,\theta)\right) \tag{0.0.34}$$

As stated in Section 2.1.1, we can use the same deformation function $d(r,\theta)$ when considering either the effective main reflector or the true main reflector.

3.2 A model of pointing errors and main reflector deformation

Since the antenna may be mispointed, and since the dish may be deformed by its own weight, thermal gradients, and other factors, we can model these errors using a dish deformation function $d(r,\theta)$, which can take not only deformations of the main reflector but also tilts of the dish, which correspond directly to pointing errors, into account. Combining equations (0.0.31) and (0.0.34) yields:

$$\begin{gathered}U_f(\rho,\phi)=\frac{\exp\left(j\frac{k}{2f}\rho^2\right)}{j\lambda f}\cdot \\ \int_0^R\int_{-\pi}^{\pi} r\exp\left(j\frac{4\pi d(r,\theta)}{\lambda}\right)\exp\left(-j\frac{2\pi}{\lambda f}r\rho\cos(\theta-\phi)\right)d\theta dr\end{gathered} \tag{0.0.35}$$

This Fourier transform relationship allows us to examine the effect of pointing and subreflector position changes on the focal plane field $U_f(\rho,\phi)$, and it is the focal plane field which determines received signal energy as it strikes the seven-horn AFCS assembly. We can decompose $d\left(r,\theta\right)$ into two components: a controllable component consisting of tilts and a defocus error (which correspond to pointing and to subreflector *z*-axis position) and an uncontrollable component consisting of all other errors. Let $d_C\left(r,\theta\right)$ denote the controllable component and $d_U\left(r,\theta\right)$ denote the uncontrollable component. We write:

$$d\left(r,\theta\right)=d_C\left(r,\theta\right)+d_U\left(r,\theta\right) \tag{0.0.36}$$

The controllable component, $d_C\left(r,\theta\right)$, may be broken down into three terms for *XEL*, *EL*, and *Z*, as follows. Define the first four Zernike polynomials given in [3]:

$$\begin{aligned} U_0^0\left(r,\theta\right)&=1\\ U_1^{-1}\left(r,\theta\right)&=r\cos\theta\\ U_1^1\left(r,\theta\right)&=r\sin\theta\\ U_2^0\left(r,\theta\right)&=2r^2-1 \end{aligned} \tag{0.0.37}$$

These functions, evaluated over the unit circle, have special properties which are discussed in detail by Born and Wolf [3]. However, we note that $U_1^{-1}\left(r,\theta\right)$ and $U_1^1\left(r,\theta\right)$ correspond directly to x-tilt, or *XEL* pointing, and y-tilt, or *EL* pointing, respectively. Furthermore, $U_2^0\left(r,\theta\right)$, in the context of physical optics, often refers to defocus. As such, it closely corresponds to subreflector *Z*-position (or, under other circumstances, to wind-induced deformation of the main reflector). We note, however, that $U_2^0\left(r,\theta\right)=2r^2-1$ both shifts the edges of the main reflector dish forward while shifting the center backward, which is not what we desire. The center of the dish is a fixed point, and we include the “piston” term $U_0^0\left(r,\theta\right)$ as an additional term to remove any shifts of the center of the dish in order to maintain a physically realistic model. By setting the coefficients of $U_0^0\left(r,\theta\right)$ and $U_2^0\left(r,\theta\right)$ both equal to a_2^0, we insure that our model of correctable deformation remains physically realistic. We may now write:

$$d_C\left(r,\theta\right)=a_1^{-1}U_1^{-1}\left(\frac{r}{R},\theta\right)+a_1^1U_1^1\left(\frac{r}{R},\theta\right)+a_2^0\left[U_2^0\left(\frac{r}{R},\theta\right)+U_0^0\left(\frac{r}{R},\theta\right)\right] \quad (0.0.38)$$

where *R* is the radius of the main reflector. The coefficients a_1^{-1}, a_1^1, and a_2^0 correspond directly to *XEL*, *EL*, and *Z*, respectively, and the appropriate linear scaling factors relating the two equivalent sets of triples will be derived in Chapter 4. Decomposing the effective aperture field function yields:

$$\begin{aligned}U_a\left(r,\theta\right)&=\exp\left(j\frac{4\pi}{\lambda}\left(d_C\left(r,\theta\right)+d_U\left(r,\theta\right)\right)\right)\\&=e^{j\frac{4\pi}{\lambda}\left(a_1^{-1}U_1^{-1}\left(\frac{r}{R},\theta\right)+a_1^1U_1^1\left(\frac{r}{R},\theta\right)+a_2^0\left(U_0^0\left(\frac{r}{R},\theta\right)+U_2^0\left(\frac{r}{R},\theta\right)\right)\right)}e^{j\frac{4\pi}{\lambda}d_U\left(r,\theta\right)}\end{aligned} \quad (0.0.39)$$

Substituting equation (0.0.39) into equation (0.0.35) gives us:

$$U_f\left(\rho,\phi\right)=\frac{e^{j\frac{k}{2f}\rho^2}}{j\lambda f}\int_0^R\int_{-\pi}^{\pi} r\exp\left(\begin{matrix}\left[-j\frac{2\pi}{\lambda f}r\rho\cos\left(\theta-\phi\right)\right]+\\ \left[j\frac{4\pi}{\lambda}\left(\begin{matrix}a_1^{-1}U_1^{-1}\left(\frac{r}{R},\theta\right)+a_1^1U_1^1\left(\frac{r}{R},\theta\right)\\ +a_2^0\left(U_0^0\left(\frac{r}{R},\theta\right)+U_2^0\left(\frac{r}{R},\theta\right)\right)\end{matrix}\right)\right]+\\ \left[j\frac{4\pi}{\lambda}d_U\left(r,\theta\right)\right]\end{matrix}\right)d\theta dr \qquad (0.0.40)$$

Equation (0.0.40) expresses the focal plane field as a function of both the uncorrectable antenna dish deformation and the correctable pointing and defocus. Since the pointing elevation θ_{EL} has a very significant effect on $d_U\left(r,\theta\right)$ [22,37,39,40,42-44], the focal plane field $U_f\left(\rho,\phi\right)$ is also heavily affected as shown in Figure 1.4 through Figure 1.6. $P_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$ will thus be heavily affected by overall pointing elevation, and the optimal operating point in *(XEL,EL,Z)*-space may vary significantly with θ_{EL}.

3.3 Observing the Received Signal Power and Focal Plane Information with the AFCS

Our objective is to maximize the total received signal power from the seven horns as a function of *XEL*, *EL*, and *Z* (or, equivalently, a_1^{-1}, a_1^1, and a_2^0). To do this, we must find the total received power *P* as a function of the focal plane field $U_f\left(\rho,\phi\right)$. Zohar and Vilnrotter treat the complex baseband outputs of the AFCS horns as samples of the focal plane field taken at the horn centers [57], and this approximation will prove very useful in our treatment of this problem as well. We will discuss some of the reasoning behind this approximation prior to applying it to the problem at hand.

3.3.1 Sampling Functions

For simplicity, we will first examine the case of rectangular sampling of a two-dimensional function. Our treatment will closely follow that of Wozencraft and Jacobs [56] and of Bracewell [4], although there exist numerous other texts containing similar material. We will argue that the horns of the AFCS act as approximate sampling functions, justifying the point-sample approximation of Zohar and Vilnrotter [57] and allowing us to come up with a way of expressing total received power as a function of the pointing and subreflector control inputs. Here, in order to simplify the treatment, we will deal with hypothetical functions in *rectangular* coordinates. Let $F\left(u,v\right)$ be the Fourier transform of a strictly bandlimited function $f\left(x,y\right)$ which satisfies:

$$F\left(u,v\right)=0 \quad \forall u,v:\left|u\right|\geq W_1,\left|v\right|\geq W_2 \tag{0.0.41}$$

Equation (0.0.41) implies that $F\left(u,v\right)$ is zero outside of a rectangular window, making the function $f\left(x,y\right)$ strictly bandlimited but strictly unlimited in spatial extent. By the sampling theorem, there exists a set of samples $\left\{f_{m,n}\right\}$ such that:

$$f\left(x,y\right)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}f_{m,n}\zeta_{m,n}\left(x,y\right) \tag{0.0.42}$$

where

$$\zeta_{m,n}(x,y)=\left(\sqrt{2W_1}\frac{\sin\left(2\pi W_1\left(x-\frac{m}{2W_1}\right)\right)}{2\pi W_1\left(x-\frac{m}{2W_1}\right)}\right)\left(\sqrt{2W_2}\frac{\sin\left(2\pi W_2\left(y-\frac{m}{2W_2}\right)\right)}{2\pi W_2\left(y-\frac{m}{2W_2}\right)}\right) \quad (0.0.43)$$

Equation (0.0.43) gives us a sampling function. The infinite set of these sampling functions $\zeta_{m,n}(x,y)$ forms a complete, orthogonal basis over the set of all bandlimited functions $f(x,y)$ whose Fourier transforms $F(u,v)$ satisfy equation (0.0.43). We can find the expansion coefficients by:

$$f_{m,n}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\zeta_{m,n}(x,y)dxdy \quad (0.0.44)$$

The function $\zeta_{m,n}(x,y)$ is called a *sampling function* [56]since it can be used to recover samples of the bandlimited function $f(x,y)$. If we expand equation (0.0.44) by substituting with equation (0.0.43), we obtain:

$$f_{m,n}=2\sqrt{W_1W_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\frac{\sin\left(2\pi W_1\left(x-\frac{m}{2W_1}\right)\right)}{2\pi W_1\left(x-\frac{m}{2W_1}\right)}\right)\left(\frac{\sin\left(2\pi W_2\left(y-\frac{m}{2W_2}\right)\right)}{2\pi W_2\left(y-\frac{m}{2W_2}\right)}\right)f(x,y)dxdy \quad (0.0.45)$$

which is really a sampled version of the following convolution:

$$f(x,y)*\xi(-x,-y) \quad (0.0.46)$$

where

$$\zeta(x,y)=2\sqrt{W_1W_2}\frac{\sin(2\pi W_1x)}{2\pi W_1x}\frac{\sin(2\pi W_2y)}{2\pi W_2y} \quad (0.0.47)$$

Since $\zeta(x,y)=\zeta(-x,-y)$, the convolution in equation (0.0.46) is really a low-pass filtering operation on $f(x,y)$, which has no effect beyond multiplication by a scalar constant since the Fourier transform of $f(x,y)$ satisfies equation (0.0.41).

Hence, sampling the convolution, which is a scaled replica of the original bandlimited function, is equivalent to sampling the function itself except for a constant scaling factor since convolution with a low-pass filter of the same bandwidth as the original function does not fundamentally change the function.

Since the horns are arranged in a hexagonal array and have circular apertures, we need to consider the case of hexagonal sampling. Once again, sampling functions do exist, and we describe Petersen and Middleton's optimal sampling method first [33]. The optimal sampling lattice is a hexagonal grid, which can also be described as a 120^0 rhombic grid [33,57]. If the function has a spectrum that is circularly limited with bandwidth *B* (described as a spectral bound of $2\pi B$ in wave number space by Petersen and Middleton [33]), the optimal sampling grid has spacing $\frac{1}{B\sqrt{3}}$ between the hexagonal sampling points. The sampling function, described in *(x,y)*-coordinates, is given by:

$$s(x,y)=\frac{\begin{pmatrix} 2x\cos\left(\frac{2\pi Bx}{\sqrt{3}}\right)\cos(2\pi By)-2x\cos\left(\frac{4\pi Bx}{\sqrt{3}}\right) \\ -2\sqrt{3}\sin\left(\frac{2\pi Bx}{\sqrt{3}}\right)\sin 2\pi By \end{pmatrix}}{(2\pi B)^2\, x\left(x^2-3y^2\right)} \tag{0.0.48}$$

The expression in equation (0.0.48) is the optimal sampling function which satisfies the orthogonality condition [33]. It is also a complicated function. In practice, one could use a simpler function in polar coordinates [4,33]:

$$s(r)=\frac{J_1(2\pi Br)}{2\sqrt{3}Br} \tag{0.0.49}$$

where B is the bandwidth of the original function and where $J_1(x)$ is a Bessel function of the first kind, order 1. It should be pointed out that the approximate, suboptimal sampling function of equation (0.0.49) violates the orthogonality condition and is not the true optimal sampling function [33] although it is far simpler due to its circular symmetry.

Both the circularly symmetric sampling function of equation (0.0.49) and the rectangular sampling function of equation (0.0.47) are spatially unlimited due to the strict frequency limitation. For this reason, the horn aperture pattern cannot be an exact sampling function since each horn aperture is a strictly spatially limited function. Nevertheless, the AFCS could be regarded as performing point sampling [57], making this approximation a useful one. From this point forward, the point sampling approximation will be used with the understanding that although the horns are good approximate sampling functions they are not perfect.

3.4 Achieving Optimal Performance

Given that the horn outputs are very nearly samples of the focal plane field, and given that equation (0.0.40) gives us the focal plane field as a function of the user inputs and of the uncorrectable part of the dish deformation, we are ready to express power as a function of four quantities: *XEL*, *EL*, *Z*, and $d_U(r,\theta)$. Equivalently, we can express power as a function of a_1^{-1}, a_1^1, a_2^0, and $d_U(r,\theta)$. Let the set of horn centers, in the polar coordinates of the focal plane, be given by:

$$\left\{(\rho_1,\phi_1),(\rho_2,\phi_2),(\rho_3,\phi_3),(\rho_4,\phi_4),(\rho_5,\phi_5),(\rho_6,\phi_6),(\rho_7,\phi_7)\right\} \tag{0.0.50}$$

The total received power is equal to the sum of the squared magnitudes of each complex baseband horn output, which is, under the point sampling assumption:

$$P = \sum_{n=1}^{7} \left| U_f\left(\rho_n,\phi_n\right) \right|^2 \tag{0.0.51}$$

When we substitute equation (0.0.40) into equation (0.0.51), we obtain:

$$P_{\theta_{EL}}\left(a_1^{-1},a_1^1,a_2^0\right) = \sum_{n=1}^{7} \left| \frac{e^{j\frac{k}{2f}\rho_n^2}}{j\lambda f} \int_0^R \int_{-\pi}^{\pi} r e^{\left[-j\frac{2\pi}{\lambda f} r\rho_n \cos\left(\theta-\phi_n\right) \right] + \left[j\frac{4\pi}{\lambda}\left(\begin{array}{l} a_1^{-1}U_1^{-1}\left(\frac{r}{R},\theta\right) + a_1^1 U_1^1\left(\frac{r}{R},\theta\right) + \\ a_2^0\left(U_0^0\left(\frac{r}{R},\theta\right) + U_2^0\left(\frac{r}{R},\theta\right) \right) + d_U\left(r,\theta\right) \end{array} \right) \right]} d\theta dr \right|^2 \tag{0.0.52}$$

Here, $d_U\left(r,\theta\right)$ represents the uncorrectable portion of the dish deformation, which is a function of pointing elevation, thermal gradients, and other factors. For a given $d_U\left(r,\theta\right)$, we seek to maximize $P_{\theta_{EL}}\left(a_1^{-1},a_1^1,a_2^0\right)$ as a function of the three controllable parameters. Ultimately, our objective is to find $\left(a_1^{-1*},a_1^{1*},a_2^{0*}\right)$ to maximize $P_{\theta_{EL}}\left(a_1^{-1},a_1^1,a_2^0\right)$ in equation (0.0.52). Although one may attempt a direct optimization approach, two entirely different methods were chosen. The first is based on an interpolated least squares algorithm that estimates $\left(a_1^{-1*},a_1^{1*},a_2^{0*}\right)$, while the second uses neural networks to estimate $\left(a_1^{-1*},a_1^{1*},a_2^{0*}\right)$.

A set of SNR predicts, which are predictions of receiver SNR under ideal conditions, are available from the DSN. These provide a way of finding the maximum received power P_{opt}, which provides a target power. Given that only three control inputs (*XEL, EL,* and *Z*) are available, some of the dish deformation will remain uncorrected, and $P\left(a_1^{-1*}, a_1^{1*}, a_2^{0*}\right) \leq P_{opt}$. Given that P_{opt} is available from the DSN, one can define:

$$\Delta P_{\theta_{EL}}\left(a_1^{-1}, a_1^{1}, a_2^{0}\right) = P_{opt} - P_{\theta_{EL}}\left(a_1^{-1}, a_1^{1}, a_2^{0}\right) \qquad (0.0.53)$$

as a strictly positive quantity to be minimized. Hence, the problem of SNR maximization may also be posed as a problem of minimizing $\Delta P_{\theta_{EL}}\left(a_1^{-1}, a_1^{1}, a_2^{0}\right)$. Due to the presence of the uncorrectable portion of the deformation $d_U\left(r,\theta\right)$, there will be a positive quantity

$$\Delta P_{\min,\theta_{EL}} = P_{opt} - P_{\theta_{EL}}\left(a_1^{-1*}, a_1^{1*}, a_2^{0*}\right) \qquad (0.0.54)$$

which forms the lower bound for equation (0.0.53). The problem to be solved may, therefore, be cast either as a problem of maximizing the SNR in equation (0.0.52) or as a problem of minimizing the difference in equation (0.0.53). Equation (0.0.54) provides the best (lowest) achievable $\Delta P_{\theta_{EL}}$, and this quantity is strictly positive due to the uncorrectable deformation $d_U\left(r,\theta\right)$.

It must be noted that the ordered triple $\left(a_1^{-1}, a_1^{1}, a_2^{0}\right)$, although related to *(XEL,EL,Z)* by linear scaling factors, is *not* the same as *(XEL,EL,Z).* A derivation of the linear scaling factor for both *XEL* and *EL* is presented in Section 4.3. In the case of the linear scaling factor from a_2^0 to *Z*, the relationship is more complicated. Here, *Z* denotes the change in position of the subreflector from its nominal position while

a_2^0 denotes a change in an expansion coefficient in equation (0.0.38), and the relationship between the two is derived in Section 4.3.2.

3.5 Optimizing the complex combining weights

In equation (0.0.51), we took the sum of the output powers of the seven receiving elements to be the total power to be maximized. The problem of coherently combining the horn outputs to produce a total output of maximum SNR remains a part of this optimization problem, and this coherent combining problem has been addressed first in the original paper by Vilnrotter and Rodemich [52] and later in [6,20,46-48,50,51,53,54]. Let the noise in each receiving element have power $2\sigma^2$ (that is, σ^2 in the in-phase channel and σ^2 in the quadrature channel). The upper bound on achievable SNR is then given by [6,20,46-48,50-54]:

$$SNR_{total} \leq \sum_{n=1}^{7} \frac{\left|V_n\right|^2}{2\sigma^2} \tag{0.0.55}$$

where V_n is the complex baseband output of the *n*<u>*th*</u> receiving horn. In this case, there exists an optimal set of combining weights $\left\{w_k\right\}$ such that the maximum SNR is achieved, and these weights are given by [6,20,46-48,50-54]:

$$w_n = \frac{V_n^*}{2\sigma^2} \tag{0.0.56}$$

By using maximum likelihood estimates of the weights given in equation (0.0.56), the SNR can be brought very close to the upper bound in equation (0.0.55) [6,20,46-48,50-54]. In fact, the original paper by Vilnrotter and Rodemich shows that combining loss can be kept below 0.1 dB [52]. This validates the concept of maximizing the sum of received power at the horns in equation (0.0.51) since this maximizes the upper bound in equation (0.0.55) . The problem of maximum likelihood estimation of coefficients to actually achieve the upper bound that equation (0.0.51) implies has been treated in the original paper [52] and in follow-on work [6,46,48,50,51,53,54], and the reader is referred there for further information since the problem of estimating the optimal offset *(XEL,EL,Z)* is the focus of this dissertation. It should be noted, however, that the optimal combining weights vary significantly as a function of overall antenna deformation and must be updated in real-time during actual operation using maximum likelihood estimation procedures[6,46,48,50-54].

4 A Constrained Least Squares Approach to SNR Maximization

In Chapter 3 we stated our goal of maximizing SNR as a function of the ordered triple *(XEL,EL,Z)*. Let us define the point in *(XEL,EL,Z)*-space at which SNR is a maximum as *(0,0,0)*. This is the optimal operating point which yields the highest possible SNR given our three control inputs. At any given moment, the antenna may be operating at a point other than the optimal point, and we seek to estimate the antenna's present operating point in *(XEL,EL,Z)* space since knowledge of this operating point will enable us to correct pointing and focus in order to maximize SNR. Our approach will consist of attempting to find a constrained least squares estimator for *(XEL,EL,Z)* which accepts the complex baseband output vector of the AFCS as its input and returns an estimate of *(XEL,EL,Z)* as its output.

The objective function, $P_{\theta_{EL}}\left(XEL, EL, Z\right)$, is dependent on overall pointing elevation θ_{EL}. Thus it is necessary to find this optimal operating point for any given θ_{EL} since, during the course of tracking, θ_{EL} will vary over a wide range as the spacecraft or other target rises and sets in the sky due to Earth's rotation. We first consider the estimation problem for a fixed θ_{EL} in Sections 4.1 through 4.4, and we briefly discuss extensions of this estimator to handle arbitrary θ_{EL} in Section 4.5. Section 4.1 is a discussion of the constraints on our least-squares problem imposed by physical reality. With those constraints given, Section 4.2 defines the least-squares problem itself. The method used to generate the data reference tables used in our estimation algorithm is given in Section 4.3. A practical algorithm, the *interpolated least-squares* estimation algorithm for estimating *(XEL,EL,Z)*, is given in Section 4.4, and a brief discussion of pointing elevation dependence is given in Section 4.5. This chapter concludes with a practical discussion of the use of estimation algorithms in real-life acquisition and tracking in Section 4.6.

4.1 Focal plane field information from the AFCS

The baseband output of the seven-element AFCS system is represented by a seven-element complex voltage vector given by:

$$\tilde{\mathbf{r}} = \begin{bmatrix} \tilde{r}_1 & \tilde{r}_2 & \dots & \tilde{r}_7 \end{bmatrix} \qquad (0.0.57)$$

In a situation in which the source and antenna remain absolutely stationary with respect to each other, in which absolute focal plane phase recovery is possible, and in which channel conditions are static, the above vector would be a vector of focal plane samples as discussed in Section 3.3. Under such conditions, one could write:

$$E\left[\tilde{r}_n\right] = U_f\left(\rho_n, \phi_n\right) \qquad (0.0.58)$$

where $\left(\rho_n, \phi_n\right)$ is the center of the *n<u>th</u>* horn. Unfortunately, due to the real-life factors discussed here, equation (0.0.58) from the previous case ***does not hold.*** In a realistic scenario, the following difficulties are present:

1. Time-varying received frequency due to the Doppler effect. This is caused by motion of the spacecraft and rotation of the Earth.

2. Changes in weather conditions and other channel conditions that result in changes in received signal power. Although SNR predicts exist at the DSN, there are factors including atmospheric effects and other effects that cause available received power to vary from the DSN predict models.

The existing system uses sky frequency predicts in conjunction with a phase-locked-loop to lock the signal. As such, it maintains the central horn, which has the highest signal power in normal operation, at zero-phase. This means that $\tilde{r}_1$ in equation (0.0.57) is kept at zero phase by this system. If we compare $\tilde{r}_1$ to $U_f\left(\rho_1,\phi_1\right)$, we see that while the focal plane field $U_f\left(\rho_1,\phi_1\right)$ at the center horn's location $\left(\rho_1,\phi_1\right)$ is, in general, a complex function (with a phase and amplitude depending on the received field), the central horn output yielded by this system is set to zero phase.

Since all phases of the outer horns are found relative to that of the central horn, it is true that *relative* phase relationships *among the horns* are preserved. However, we do not have the absolute phase of $U_f\left(\rho_1,\phi_1\right)$ available to us since the phase-locked loop operates to keep the center channel's output at zero-phase. This prevents complete reconstruction of the focal plane field and its absolute phase information via the AFCS. Furthermore, random variations in spacecraft power and channel conditions result in random amplitude of the received signal [47]. Hence, even the magnitude of $\tilde{r}_1$ could be different from the magnitude of $U_f\left(\rho_1,\phi_1\right)$ that would be observed under ideal conditions.

For a given operating point *(XEL,EL,Z)*, random changes in absolute signal amplitude imply that we now lack a unique one-to-one mapping between the operating point *(XEL,EL,Z)* and the received vector $\tilde{\mathbf{r}}$. Under noiseless conditions the relative phases among the horns would remain the same, but the amplitudes of the channels would only maintain their relative relationships while the absolute amplitude would be a random variable. Hence, the problem at hand is constrained by both the lack of absolute focal plane field phase knowledge and by the randomness of the amplitude of the central channel's output. This makes our problem a constrained one, and in Section 4.2 we will see it is a least-squares problem.

This situation is handled in the following manner. Define:

$$\tilde{\mathbf{x}} \triangleq \frac{\tilde{\mathbf{r}}}{\tilde{r}_1} \tag{0.0.59}$$

In equation (0.0.59), we have normalized all seven complex baseband outputs by dividing by the central horn's complex baseband output, which is a real output due to the fact that the central horn has zero-phase as a result of the operation of the phase-locked loop. Let $\tilde{x}_n$ be defined as the <u>*nth*</u> element of the vector $\tilde{\mathbf{x}}$. We always have:

$$\tilde{x}_1 = 1 \tag{0.0.60}$$

Hence, there is no information in the first element of $\tilde{\mathbf{x}}$. We can drop the first element and define the vector $\tilde{\mathbf{y}}$ as a six-element complex vector as follows:

$$\begin{aligned} \tilde{\mathbf{y}} &= \left[\tilde{x}_2 \quad \tilde{x}_3 \quad \dots \quad \tilde{x}_7\right] \\ &= \left[\tilde{y}_1 \quad \tilde{y}_2 \qquad \tilde{y}_6\right] \\ \text{where } \tilde{y}_n &= \tilde{x}_{n+1} \quad n = 1, 2, \dots, 6 \end{aligned} \tag{0.0.61}$$

The above vector is a new complex voltage vector consisting of the normalized outer horn complex voltage outputs, where normalization is performed by dividing by the central horn's complex voltage output. Given an operating point *(XEL,EL,Z)*, we obtain the vector $\tilde{\mathbf{r}}$ which is corrupted by AWGN. We then obtain the vector $\tilde{\mathbf{y}}$ by dividing the outer horn outputs by the center horn's output at complex baseband, thus obtaining the six complex numbers that comprise vector $\tilde{\mathbf{y}}$. We can, equivalently, define a 12-element real vector $\mathbf{y}$ whose elements are the real and imaginary parts of the elements of $\tilde{\mathbf{y}}$ as follows:

$$\begin{aligned} y_{2n-1} &= \mathrm{Re}\left[\tilde{y}_n\right] \\ y_{2n} &= \mathrm{Im}\left[\tilde{y}_n\right] \\ \mathbf{y} &= \left[\mathrm{Re}\left[\tilde{y}_1\right] \quad \mathrm{Im}\left[\tilde{y}_1\right] \quad \dots \quad \mathrm{Re}\left[\tilde{y}_6\right] \quad \mathrm{Im}\left[\tilde{y}_6\right]\right] \\ &= \left[y_1 \quad y_2 \quad \dots \quad y_{12}\right] \end{aligned} \tag{0.0.62}$$

For a given operating point *(XEL,EL,Z)* at pointing elevation θ_{EL}, we define:

$$\vec{\eta}_{\theta_{EL}}\left(XEL, EL, Z\right) \triangleq E\left[\mathbf{y}\right] \tag{0.0.63}$$

which is the expected value of $\mathbf{y}$ at the given operating point *(XEL,EL,Z)* at elevation θ_{EL}.

4.2 The constrained least squares problem

Ultimately, we seek an estimate of *(XEL,EL,Z)* that we can use to correct the pointing and subreflector position in an effort to bring the system to the optimal point

(0,0,0) where SNR is maximized. One approach to this problem is that of finding the ordered triple $\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right)$ that minimizes the following function:

$$g_{\theta_{EL}}\left(XEL, EL, Z\right) = \left\|\mathbf{y} - \vec{\eta}_{\theta_{EL}}\left(XEL, EL, Z\right)\right\|^{2} \qquad (0.0.64)$$

at the current pointing angle θ_{EL}. Here, $\mathbf{y}$ is the 12-element vector defined in equation (0.0.62) that corresponds to received vector $\tilde{\mathbf{r}}$. In the noiseless case, the ordered triple $\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right)$ that achieves this minimum will give us $g_{\theta_{EL}}\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right) = 0$ since, in the absolutely noiseless case, $\mathbf{y}$ is a purely deterministic function of θ_{EL} and of the operating point *(XEL,EL,Z)* and would thus be exactly equal to $\vec{\eta}_{\theta_{EL}}\left(XEL, EL, Z\right)$. In that case, the ordered triple $\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right)$ for which $g_{\theta_{EL}}\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right) = 0$ would be exactly the current operating point. In practice, however, $\mathbf{y}$ always contains some noise because the vector of AFCS outputs $\tilde{\mathbf{r}}$ is corrupted by AWGN, and our objective is to find $\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right)$ to minimize $g_{\theta_{EL}}\left(XEL, EL, Z\right) = \left\|\mathbf{y} - \vec{\eta}\left(XEL, EL, Z\right)\right\|^{2}$. This is a constrained least squares problem. The algorithms that we will use to obtain $\left(\widehat{XEL}, \widehat{EL}, \widehat{Z}\right)$ are designed to solve this constrained least squares problem, and these algorithms are called *interpolated least squares* algorithms. They are so-called because we are solving a constrained least-squares problem and because they use interpolation to assist in finding the minimum of $g_{\theta_{EL}}\left(XEL, EL, Z\right) = \left\|\mathbf{y} - \vec{\eta}_{\theta_{EL}}\left(XEL, EL, Z\right)\right\|^{2}$. In Section 4.3

we discuss the problem of obtaining the vectors $\vec{\eta}_{\theta_{EL}}(XEL, EL, Z)$ for use in a lookup table. In Section 4.4, we provide a practical implementation of an algorithm to find the minimum of $g_{\theta_{EL}}(XEL, EL, Z)$ as a function of *(XEL,EL,Z)* using quadratic interpolation.

4.3 Obtaining $\vec{\eta}_{\theta_{EL}}(XEL, EL, Z)$

The basic procedure for obtaining $\vec{\eta}_{\theta_{EL}}(XEL, EL, Z)$ is that of performing a raster scan. In a raster scan, the antenna is swept through a rectangular grid of points in *(XEL,EL,Z)*-space similar to that of Figure 4.1.

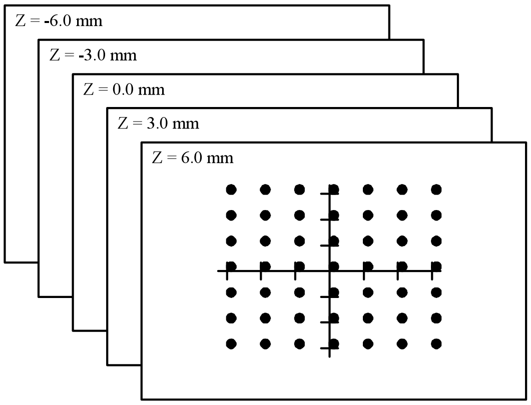

Figure 4.1: Illustration of a three-dimensional data set.

Since, in real life, incoming data are corrupted by noise, it is necessary to minimize noise effects in the data set by taking several vector samples at each point in *(XEL,EL,Z)* space. Since these samples are taken over a period not exceeding ten seconds, the resulting average is unaffected by changes in received spacecraft power at the given point in *(XEL,EL,Z)*-space since those changes occur on a slower time scale in practice. The resulting averaged vector $\tilde{\mathbf{r}}$ is then used to compute the estimate of the mean vector $\vec{\eta}_{\theta_{EL}}\left(XEL, EL, Z\right)$, which is stored in a lookup table for future reference. In the example of Figure 4.1, the raster scan is taken at 5 values of subreflector defocus parameter *Z*: -6.0 mm, -3.0 mm, 0.0 mm, 3.0 mm, and 6.0 mm. In each "defocus plane" shown in Figure 4.1, which is labeled with a given value of *Z*, the antenna is swept through *(XEL,EL)*-space from -3.0-millidegrees to +3.0-millidegrees in both pointing directions in 1.0-millidegree steps. The raster scan of Figure 4.1 gathers a total of 245 reference data vectors which are estimates of $\vec{\eta}_{\theta_{EL}}\left(XEL, EL, Z\right)$ for each of the points in the scan.

4.3.1 Translating between *(XEL,EL)* and tilt coefficients

In this section, we will seek to understand the effect of a pointing offset in *(XEL,EL)* in terms of movement of the focal plane field. A change in pointing can be modeled by a tilt of the main reflector, and such tilts are modeled by:

$$U_f\left(\rho,\phi\right)=\int_0^{\infty}\int_{-\pi}^{\pi}U_a\left(r,\theta\right)e^{j\left(\frac{4\pi}{\lambda}\left(a_1^{-1}U_1^{-1}\left(\frac{r}{R},\theta\right)+a_1^1U_1^1\left(\frac{r}{R},\theta\right)\right)\right)}e^{-j\left(\frac{2\pi}{\lambda f}r\rho\cos(\theta-\phi)\right)}rd\theta dr \quad (0.0.65)$$

where $U_a\left(r,\theta\right)$ is an arbitrary aperture plane field. We have assumed that the circular aperture cutoff is part of the aperture field function $U_a\left(r,\theta\right)$, leading to the infinite integration limit. The treatment will be simpler if we switch to rectangular coordinates. Rewriting some of the terms of equation (0.0.38) in rectangular coordinates, we have:

$$\begin{aligned} U_1^{-1}\left(x,y\right)&=x \\ U_1^{1}\left(x,y\right)&=y \end{aligned} \quad (0.0.66)$$

Re-writing equation (0.0.65) in rectangular coordinates, and assuming that the circular aperture limit is already taken into account in the aperture field function itself, we have:

$$\begin{aligned} U_f(u,v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_a(x,y) e^{j\left(\frac{4\pi}{\lambda}\right)\left(a_1^{-1}\frac{x}{R}+a_1^1\frac{y}{R}\right)} e^{-j\left(\frac{2\pi}{\lambda f}\right)(xu+yv)} dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_a(x,y) e^{-j\left(\frac{2\pi}{\lambda f}\right)\left(x\left(u-2\frac{f\,a_1^{-1}}{R}\right)+y\left(v-2\frac{f\,a_1^1}{R}\right)\right)} dxdy \end{aligned} \qquad (0.0.67)$$

Hence, an *x*-tilt coefficient of a_1^{-1} will yield a translation in the focal plane of $2\frac{f\,a_1^{-1}}{R}$ and similarly for a *y*-tilt.

We translate tilts into units of millidegrees with the objective of finding how many millimeters of translation we get in the focal plane for each millidegree of dish tilt. From Section 3.1, we note that a change in the coefficients a_1^1 and a_1^{-1} must be doubled to get the true movement in the dish edge because we have folded these coefficients into the deformation function, which is phased-doubled. We first translate from millidegrees of dish tilt to millimeters in coefficients using the small angle approximation:

$$1\,m\deg \frac{1\deg}{1000\,m\deg}\frac{\pi\, rad}{180\deg}35\,meters\frac{1000\,mm}{1\,meter}\frac{1}{2} = 0.3054\,mm \qquad (0.0.68)$$

The coefficient changes by 0.3054-millimeters for each millidegree of tilt, using the small angle approximation. Multiplying the tilt coefficient by $2\frac{f}{R} \approx 8.6857$ yields

2.6526-millimeters of focal plane translation for each one-millidegree change in pointing.

We next consider the problem of determining a sampling interval in *(XEL,EL)*-space. The following issues need to be considered first:

1. The spatial frequency content of the objective function $g_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$ is unknown. The normalized input vector $\mathbf{y}$ defined in equation (0.0.62) and the vectors $\vec{\eta}_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$ are not vectors containing samples of the focal plane field: they contain processed focal plane field information instead of actual samples.
2. The elements of the mean normalized voltage vector $\vec{\eta}_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$ do not have the same frequency spectrum as the focal plane field function $U_f\left(\rho,\phi\right)$. The very act of dividing the outer horn voltages by that of the central horn creates a mapping from $\mathbb{C}^7$ to $\mathbb{C}^6$.

The statements above imply that any sampling interval calculated for $U_f\left(\rho,\phi\right)$ in *(XEL,EL,Z)*-space will ***not***, in general, be applicable to any one of the components of the vector function $\vec{\eta}_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$. If a Nyquist interval is calculated for $U_f\left(\rho,\phi\right)$ (based on the bandwidth limitation imposed by the circular aperture) then, in general, that Nyquist interval is not applicable to the vector function $\vec{\eta}_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$ or to the scalar function $g_{\theta_{EL}}\left(XEL,\ EL,\ Z\right)$. An example from one-dimensional signal processing illustrates this concept. Given a time-domain signal *x(t)* with bandwidth *B*, the Nyquist rate is *2B*. If we consider the spectrum of the signal $\left|x(t)\right|^2$, we have a signal bandwidth of *2B* and a Nyquist rate of *4B*, so in this case processing has doubled its bandwidth. If we pass *x(t)* through a low-pass filter of bandwidth *B/2*, then the new signal's bandwidth is halved and so is the Nyquist rate. Processing can alter the frequency characteristics of signals, and since the computation of $g_{\theta_{EL}}(XEL, EL, Z)$ involves significant non-linear processing of the focal plane field, the frequency content is not known due to the translation from $\mathbb{C}^7$ to $\mathbb{C}^6$. In general, although the circular aperture limitation does impose a bandwidth limit on the focal plane field, this does not translate readily into a bandwidth limit on either the vector function $\vec{\eta}_{\theta_{EL}}(XEL, EL, Z)$ or on the objective function to be minimized, which is $g_{\theta_{EL}}(XEL, EL, Z)$. Hence, we do not attempt to use the bandwidth limitation on $U_f(\rho,\phi)$ imposed by the circular aperture to determine a sampling interval for $g_{\theta_{EL}}(XEL, EL, Z)$.

Our choice of spacing in the raster scan is governed by practical considerations. Since the DSN's goal is to keep pointing errors below 0.8-millidegrees, a 1.0-millidegree interval is typically chosen for the raster scan in the *XEL* and *EL* directions. Such an interval choice is determined not only by the DSN's goal but also by the fact that Vilnrotter and Fort, in their paper discussing experiments conducted in 1999, state that the system is able to resolve errors of 1.0-millidegrees or less [47] and by the fact that Zohar and Vilnrotter have shown that, in principle, the AFCS is capable of fine pointing discrimination with errors of 1.0-millidegree or less as well [57]. The choice of spacing in *Z*, the subreflector's *z*-position offset, is also governed by practical considerations. Due to the time it takes to complete a raster scan and due to changes in θ_{EL} over time resulting from the Earth's rotation, a compromise is made between resolution in *Z* and the need to complete a raster scan without very large changes in θ_{EL} occurring during the scan period. In the real world experiments at DSS-13, 2.0-millimeter spacing in *Z* was chosen.

4.3.2 The defocus coefficient

In the previous subsection we derived the coefficients relating a_1^{-1} and a_1^1 to *XEL* and *EL*. We did not derive such a coefficient relating *Z*, the subreflector position, to a_2^0, the parabolic coefficient in the deformation function. This is carried out here.

Using equation (0.0.2) to find $\frac{\partial f_p}{\partial z}$ we can accurately approximate the change in focal length Δf_p corresponding to dish edge movement Δz. If the dish is deformed by a purely parabolic deformation, such that its new shape is also parabolic, the result is a change in focal length. Define Δz to be the change in the *z*-position of the dish edge due to a purely parabolic deformation. Define Δf_p to be the resulting change in true focal length. We can approximate Δf_p accurately for small Δz as follows:

$$\begin{aligned}\Delta z &\approx \frac{\partial z}{\partial f_p}\Delta f_p \\ &= -\frac{R^2}{4{f_p}^2}\Delta f_p\end{aligned} \tag{0.0.69}$$

where *R* is the radius of the dish (35-meters) and f_p is the true focal length (28.89-meters). Each +1.0-millimeter change in Δf_p yields a -0.3669-millimeter change in Δz. We can also say that each +1.0-millimeter in Δz yields -2.7253-millimeters in Δf_p.

A purely parabolic deformation of the main reflector thus causes a shift in its focal point. According to Jamnejad [25], one can compensate for a change in the focal point by moving the subreflector. Define ΔS to be the change in subreflector position along the *z*-axis. This change is magnification dependent. Using our model's effective focal length of 152-meters and the parabolic reflector's focal length of 28.89-meters [32], we obtain:

$$M = \frac{f}{f_p} = \frac{152\,meters}{28.89\,meters} \approx 5.26 \tag{0.0.70}$$

This leads to [25]:

$$\begin{aligned}\Delta S &= \frac{M^2}{M^2+1}\Delta f_p \\ &\approx 0.9651\Delta f_p\end{aligned} \tag{0.0.71}$$

This implies 2.630-millimeters of compensating subreflector movement for each millimeter of movement of the physical dish edge due to parabolic (defocus) deformation. Since, in real life, dish edge movements from a parabolic deformation do not exceed 3-millimeters [37,44], the maximum amount of subreflector movement required to compensate is roughly 7.89-millimeters. The subreflector can move between -11.43 cm to +10.668 cm in the *z*-direction [1], and this is a much greater degree of travel than what would be required for compensating for any defocus deformation that would occur in the real world.

Define *P* as the number of millimeters of dish edge movement due solely to parabolic distortion. We now supply a series of conversion factors needed to convert among a_2^0, *Z*, and *P* for pure defocus deformations for the 70-meter antenna.

1. 1.0-millimeters in a_2^0 => 2.0-millimeters in *P*
2. 1.0-millimeters in *P* => 0.5 millimeters in a_2^0
3. 1.0-millimeters in a_2^0 => 5.26-millimeters in *Z* in the opposite direction
4. 1.0-millimeters in *Z* => 0.190-millimeters of a_2^0 in the opposite direction
5. 1.0-millimeters in *P* => 2.63-millimeters in *Z* in the opposite direction.
6. 1.0-millimeters in *Z* => 0.380-millimeters in *P* in the opposite direction.

Although focal error simulations were based on the 70-meter antenna, focal error experiments were performed on the 34-meter antenna at DSS-13. It is necessary to find a set of conversion factors similar to that above for the 34-meter antenna as well. The true focal length of the main reflector at DSS-13 is 11.68-meters [32]. By equation (0.0.69), a 1.0-millimeter change Δf in focus causes a 0.5296-millimeter change in the parabolic deformation parameter *P* in the opposite direction. By equation (0.0.71), we can approximate ΔS using Δf. The conversions for the 34-meter antenna are given below.

1. 1.0-millimeters in a_2^0 => 2.0-millimeters in *P*
2. 1.0-millimeters in *P* => 0.5 millimeters in a_2^0
3. 1.0-millimeters in a_2^0 => 3.78-millimeters in *Z* in the opposite direction
4. 1.0-millimeters in *Z* => 0.265-millimeters of a_2^0 in the opposite direction
5. 1.0-millimeters in *P* => 1.89-millimeters in *Z* in the opposite direction.
6. 1.0-millimeters in *Z* => 0.530-millimeters in *P* in the opposite direction.

A movement of the subreflector does not result in a pure quadratic phase error even though we have chosen to model it as such using equations from [25] to compute equivalent subreflector movements corresponding to focal length changes. There are several circularly symmetric Zernike polynomials involved in expanding the effects of a subreflector position change, and this model makes an important simplifying assumption by treating subreflector *Z*-position changes as changes in the quadratic phase of the effective aperture plane field.

In this dissertation, we will use the parabolic deformation parameter *P*, which measures the movement of the edge of the antenna in the *z*-direction due to parabolic deformation, extensively. This parameter will be the parameter we focus on in the simulation results of Chapter 6 because the architecture of the simulation software emphasizes deformations of the main reflector. For this reason, we will refer frequently to *(XEL,EL,P)*-space instead of *(XEL,EL,Z)*-space in Chapter 6 in discussing simulation results. However, since we have no direct control of *P* in real life, the parameter *Z* (subreflector position) will be the focus of the real-world experiments in Chapter 7. This distinction is emphasized here and will be repeated again in Chapters 6 and 7. Since the phase changes caused by a change in *Z* are not purely parabolic even though we have modeled them as such, it is important to keep that distinction in mind when examining simulation results and real-world experimental results. Hence, there is an important difference between simulation and experiment that must be noted.

4.4 A practical implementation

The implementation used in practice is based on minimization of $g_{\theta_{EL}}\left(XEL, EL, Z\right)$, using fully normalized complex voltages as described in Sections 4.1 and 4.2. It is also based on the concept of Taylor series approximations and on quadratic approximation in particular.

Suppose we seek to minimize a function $w = h\left(x\right)$, and we know that the minimum value lies near $x = x_0$. We can expand this function as:

$$\begin{aligned} w = h\left(x\right) &= \sum_{n=0}^{\infty} \frac{h^{(n)}\left(x_0\right)}{n!}\left(x - x_0\right)^n \\ &= h\left(x_0\right) + h'\left(x_0\right)\left(x - x_0\right) + \frac{1}{2}h''\left(x_0\right)\left(x - x_0\right)^2 + o\left(x\right) \end{aligned} \qquad (0.0.72)$$

The term $o\left(x\right)$ will be close to zero for *x* sufficiently close to x_0 and, in that case, may be safely ignored. As long as we are sufficiently close to x_0, we may approximate $w = h\left(x\right)$ by a quadratic by ignoring $o\left(x\right)$ in equation (0.0.72). Finding the global minimum of this quadratic will yield an approximate minimum of $w = h(x)$ provided that the minimum point is sufficiently close to $x = x_0$.

Given the difficulty in determining the spatial frequency spectrum of $g_{\theta_{EL}}(XEL, EL, Z)$ and given the fact that interpolating with methods similar to that of equation (0.0.42) can be a computationally intensive process for large lookup tables the following procedure is used in practice:

1. Compute $w = g_{\theta_{EL}}(XEL, EL, Z) = \left\|\mathbf{y} - \vec{\eta}_{\theta_{EL}}(XEL_n, EL_n, Z_n)\right\|^2$ for each raster scan point (XEL_n, EL_n, Z_n) in the reference table. Choose the point (XEL_0, EL_0, Z_0) from the lookup table that minimizes *w*. This is the starting point, and $w_0 = \left\|\mathbf{y} - \vec{\eta}_{\theta_{EL}}(XEL_0, EL_0, Z_0)\right\|^2$ is the lowest value of *w* over the points contained in the lookup table.
2. Let ΔXEL denote the spacing in *XEL* in this table. We use the adjacent values $w_{-1} = \left\|\mathbf{y} - \vec{\eta}_{\theta_{EL}}(XEL_0 - \Delta XEL, EL_0, Z_0)\right\|^2$ and $w_1 = \left\|\mathbf{y} - \vec{\eta}_{\theta_{EL}}(XEL_0 + \Delta XEL, EL_0, Z_0)\right\|^2$ in order to determine a quadratic.
3. Fit a parabola to the set of points $\{(XEL_0 - \Delta XEL, w_{-1}), (XEL_0, w_0), (XEL_0 + \Delta XEL, w_1)\}$.
4. Find the value $\widehat{XEL}$ which minimizes this parabola. It is our estimate of *XEL*.
5. Repeat the same process for both *EL* and *Z*.

The above procedure is not the optimal quadratic interpolation procedure: a more sophisticated procedure seeks a best-fit quadratic in all three variables and seeks to minimize that quadratic. In practice, however, the procedure above has yielded results which are very close to those obtained with an optimal quadratic fit but with reduced computational complexity, and the one-dimensional interpolation procedure was the one used in real-life experiments. Other interpolation methods, including sinc-function and cubic-spline interpolation, are also possible and are tested in the simulations of Chapter 6.

This interpolated least squares algorithm serves as our performance baseline against which neural networks are compared. It is an interpolated version of a least-squares pointing algorithm written and tested by Vilnrotter and Fort [47]. This algorithm estimates the current offset *(XEL,EL,Z)* by attempting to minimize the objective function $g_{\theta_{EL}}(XEL, EL, Z)$, so it tries to solve a least-squares minimization problem under the constraint that the six complex voltages it uses are normalized by the center horn, making this a constrained least squares problem. This algorithm suffers from two disadvantages compared to neural networks:

1. The antenna changes over time, and these changes result in a need to update the reference table, which serves as a model for this algorithm. The generation of a three-dimensional lookup table is a time-consuming task, and minimizing DSN antenna downtime is very important.
2. This algorithm must compare the received voltage vector $\mathbf{y}$ to every reference vector $\vec{\eta}_{\theta_{EL}}(XEL, EL, Z)$ in the reference table. In order to minimize the finite

window effects which result from using finite-sized lookup tables (assuming that optimal interpolation of the type illustrated in equation (0.0.42) is used) it is necessary to have a large table, and large tables are typically necessary given the wide range of antenna operating conditions. For example, a table could span a range from -3 to +3 millidegrees in both *XEL* and *EL* as well as a range of -6 to +6-millimeters in *Z* with 1-millidegree spacing in both *XEL* and *EL* and 2-millimeter spacing in *Z*, but this results in a table with 343 entries, resulting in 343 Euclidean distance computations for each incoming voltage vector from the AFCS baseband assembly during operation.

4.5 Estimation over a broad range of elevations

Since $g_{\theta_{EL}}(XEL, EL, Z)$ is strongly dependent on θ_{EL}, a method for estimating $g_{\theta_{EL}}(XEL, EL, Z)$ for all elevations is needed. We recall that knowledge of $d(r,\theta) = d_C(r,\theta) + d_U(r,\theta)$ gives us knowledge of $g_{\theta_{EL}}(XEL, EL, Z)$. Rochblatt, et al. have shown that one can compute $d(r,\theta)$ for all θ_{EL} provided that one knows $d(r,\theta)$ for at least three elevation angles and have actually computed an all-angles deformation model of the main reflector of the 70-meter antenna using this method [38,44]. Using this knowledge, one can compute the resulting focal plane field and subsequently create a model of $g_{\theta_{EL}}(XEL, EL, Z)$. Such models allow focal plane fields to be computed in principle, but this approach is not practical. The computation of focal plane fields for a given elevation θ_{EL} is not feasible in real time.

A practical method of overcoming this limitation is to create three-dimensional raster scan sets at several well-chosen values of θ_{EL}. Since main reflector deformation is a continuous function of θ_{EL} [38,44], we can safely use a raster scan set generated for a given θ_{EL} for any elevations that are close to it. In reality, raster scan sets cover a range of elevations since θ_{EL} is changing during the raster scan process, particularly in the case of three-dimensional pointing and defocus scans, but the fact that antenna deformation is a continuous function of θ_{EL} implies that for sufficiently small elevation spreads during the scanning process this method will produce valid tables. The most common method used in real-life is that of taking raster scan sets consecutively as the antenna either rises or falls in elevation, and such a technique has produced excellent results when the reference tables are applied in real-life. One needs only to select the reference table that was gathered over a range of elevations that includes the current elevation, and this set will work for the current elevation as confirmed in Chapter 7 and in [31]. From this point forward, we will always assume that we are using a three-dimensional lookup table appropriate to our current antenna elevation.

4.6 A practical note on the use of interpolated least squares estimators for pointing and focus correction

Algorithms such as interpolated least squares and neural networks will compute a correction to be applied to the antenna. In order to compensate for possible estimation errors, only a fraction of this correction is applied. Hence, it normally takes several steps to get to the optimal operating point. It is often sufficient for the algorithm to estimate the direction of the correction to be applied. Even if the magnitude is wrong, as long as the magnitude is not significantly greater than the true magnitude of correction necessary, the algorithm will approach the optimal point, defined as (0,0,0).

An example best illustrates this. Suppose the antenna starts at the point (3,-3,3) in *(XEL,EL,Z)*-space. In this case, suppose the estimator works ideally, returning the exact operating point in *(XEL,EL,Z)*-space at each iteration. Clearly, if the full correction is applied, the system will correct its pointing and focus in only one step. Suppose instead that at a given time the correction is reduced by a factor of 0.7 prior to being applied. The starting portion of a time series of corrections is shown below.

Table 1: Time series with 70% correction applied at each iteration and perfect estimation.

Time Step	Actual Operating Point	Estimated Operating Point
1	(3,-3,3)	(3,-3,3)
2	(0.9,-0.9,0.9)	(0.9,-0.9,0.9)
3	(0.27,-0.27,0.27)	(0.27,-0.27,0.27)
4	(0.081,-0.081,0.081)	(0.081,-0.081,0.081)
5	(0.00243,-0.00243,0.00243)	(0.00243,-0.00243,0.00243)

Next, suppose our estimator consistently overestimates. For simplicity, we will assume it always overestimates by a constant factor of $\frac{4}{3}$. At the point (3,-3,3) it would estimate (4,-4,4). Once again, just 70% of the correction is applied each time.

Table 2: Time series with 70% correction applied at each iteration and estimates that are 33.33% too high.

Time Step	Actual Operating Point	Estimated Operating point
1	(3,-3,3)	(4,-4,4)
2	(0.2,-0.2,0.2)	(0.267,-0.267,0.267)
3	(0.0133,-0.0133,0.0133)	(0.0178,-0.0178,0.0178)

4	(0.0009,-0.0009,0.0009)	(0.0012,-0.0012,0.0012)

We also consider the case where the estimates are consistently 50% too low.

Table 3: Time series with 70% correction applied at each iteration and estimates that are 50% too low.

Time Step	Actual Operating Point	Estimated Operating Point
1	(3,-3.3)	(1.5,-1.5,1.5)
2	(1.95,-1.95,1.95)	(0.9750,-0.9750,0.9750)
3	(1.2675,-1.2675,1.2675)	(0.6338,-0.6338,0.6338)
4	(0.8239,-0.8239,0.8239)	(0.4119,-0.4119,0.4119)
5	(0.5355,-0.5355,0.5355)	(0.2677,-0.2677,0.2677)
6	(0.3481,-0.3481,0.3481)	(0.1741,-0.1741,0.1741)
7	(0.2263,-0.2263,0.2263)	(0.1132,-0.1132,0.1132)

The important point of the hypothetical data shown from Table 1 through Table 3 is this: even if there are errors in the estimator, the use of a multiplicative factor to scale the corrections will result in more robust, reliable performance. This is especially true if there is a consistent tendency to overestimate, provided that the percentage overestimation does not cause unstable oscillations. Although this results in slower convergence in the underestimation case of Table 3, it also helps to stabilize the system and to prevent oscillations in some overestimation cases.

There are two distinct phases of operation in a practical application: *acquisition* and *tracking*. In acquisition, the antenna is several millidegrees off-point or several millimeters out of focus. In this case, the goal is to correct the error and to bring the antenna to the optimal operating point. Once this is achieved, tracking involves keep the antenna on point, and tracking accuracy is critically affected by estimation accuracy near the origin. Tracking accuracy focuses on maintaining robust performance in response to relatively small perturbations. This distinction will be important in the discussions of simulation and experimental results in Chapters 6 and 7.

5 Neural Networks: An Efficient, Adaptive Approach

The interpolated least squares algorithm is based on a constrained least-squares method for antenna pointing and subreflector defocus estimation. This algorithm is a reference against which other candidate algorithms may be compared. One of its most significant weaknesses is complexity in both time and memory. Another limitation is the lack of adaptability. If there are significant long-term changes in the shape of the main reflector due to aging, seismic damage, or other factors, new raster scans must be gathered. This is a time-consuming operation and while better and faster raster scan procedures exist [36] and have been used extensively in holography [39-41,43], we seek algorithms which can adapt to long-term changes in the antenna in real-time or in near-real time. Since time on an operational DSN antenna is always at a premium, we seek to avoid taking new raster scan sets more often than necessary.

One approach to real-time learning, which also offers significant computational savings in both time and memory compared to the interpolated least squares algorithm, is that of neural networks [16,17]. Neural networks can be trained using the same raster-scan-based reference sets used by the interpolated least squares algorithm. Radial basis function (RBF) networks may be trained using the OLS algorithm [5,17], which has proven very successful in practice in real-world experiments at DSS-14 [31] and in simulations [28-31], or multilayer feedforward networks can be trained using backpropagation or one of its many variants [16,17].

The later approach has worked well at DSS-13. Both approaches will be considered here.

5.1 The computational complexity advantage of neural networks

Starting with a complexity analysis of the interpolated least-squares algorithm, assume a reference table of size $N_1 \times N_2 \times N_3$. Since the closest point is found by searching over $N_1 N_2 N_3$ points, it is necessary to compute a squared Euclidean distance between two 12-dimensional vectors $N_1 N_2 N_3$ times each time the algorithm is run. Each Euclidean distance comparison would involve 23 additions (actually12 subtractions and 11 additions) and 12 multiplications. The algorithm's complexity is proportional to $N_1 N_2 N_3$, which is the size of the main table. At DSS-13, the lookup table ranged from -4.0 to +4.0 millidegrees in both *XEL* and *EL* in 2.0-millidegree steps, resulting in 25-point *(XEL,EL)* planes in the table. Since the defocus planes ranged from Z = -6.0-millimeters to Z=+6.0-millimeters in 2.0-millimeter steps, the table contained 175 lookup points. Hence, a total of 175 Euclidean distance computations were needed for each step. At DSS-14, where only pointing was treated, 49-point raster scans with both *XEL* and *EL* ranging from -3.0- to +3.0-millimeters in 1.0-millidegree steps were used. There were 49 Euclidean distance computations for each step.

A multilayer feedforward neural network with a single hidden layer is computationally simpler. Such a network, with 30 hidden layer neurons and 1 output node, requires the multiplication of a 12-element input vector (written as a column vector) with a 30 x 12 element synaptic weight matrix. This involves 360 multiplications and 330 additions to produce 30 neural activation levels, each of which must be added to a bias weight for another 30 additions. Each of these goes through a hyperbolic tangent function, requiring 30 calls to "tanh". Finally, these are multiplied by a 1 x 30 synaptic weight matrix resulting in one output, which is added to a bias weight. This involves 30 multiplications and 30 additions. If three networks are used, one for each of the coordinates in *(XEL,EL,Z)*, we multiply the total number of operations by 3 to obtain the computational complexity of a neural network solution. A different set of computations would be used if we were to run a single three-output network with 30 hidden layer neurons.

Table 4: Complexity analysis of three single-output multilayer feedforward networks

Phase	Additions	Multiplications	Other	Total
Matrix multiplication	330	360	0	690
Bias weight addition	30			30
Subtotal	360	360	0	720
Hyperbolic tangent			30	30
Subtotal	360	360	30	750
Matrix multiplication	29	30	0	59
Bias weight addition	1			1
Subtotal	390	390	30	810
x 3 networks	1170	1170	90	2430

Table 5: Complexity analysis of a single three-output multilayer feedforward network

Phase	Additions	Multiplications	Other	Total
Matrix multiplication	330	360	0	690
Bias weight addition	30			30
Subtotal	360	360	0	720
Hyperbolic tangent			30	30
Subtotal	360	360	30	750
Matrix multiplication	87	90	0	177
Bias weight addition	3			3
Totals	450	450	30	930

Complexities of the multilayer feedforward network and the interpolated least squares algorithm at DSS-13 are compared below (note that three single-output networks were tested at DSS-13 as time constraints prevented testing of a single three-output network):

Table 6: Complexity comparison for pointing and defocus estimation

ALGORITHM	ADDITIONS	MULTIPLICATIONS	OTHER	TOTAL
Interpolated least squares	4025	2100	0	6125
Three single-output multilayer feedforward neural networks	1170	1170	90	2430
One three-output multilayer feedforward neural network (tested only in	450	450	30	930

simulations)				

The complexity of evaluating the hyperbolic tangent function is dependent on the instruction set architecture of the underlying processor. The highly popular Intel x86 series of processors includes the hyperbolic tangent function built into the processor architecture, making evaluations relatively swift. This can result in the multilayer feedforward network yielding run times two to six times faster than that of the interpolated least squares algorithm (depending on how many networks are used) and, as shown in Chapter 7, neural networks yield real-life estimation accuracy which is virtually indistinguishable from that of the interpolated least-squares algorithm.

An example taken from DSS-14 also illustrates a similar advantage if RBF neural networks are used. At DSS-14, all experiments focused on pointing only. Each lookup table, which had both *XEL* and *EL* ranging from -3.0 to +3.0-millidegrees in 1.0-millidegree steps, contained 49 points. The complexities of the interpolated least-squares and RBF neural network algorithms once again show an advantage with neural networks. Here, two separate networks (one for *XEL* and one for *EL*) were used, with between four and eight radial basis units each. A radial basis unit must perform the following computations:

1. 23 additions and 12 multiplications (squared Euclidean distance)
2. Multiplication by the spread factor b^2.
3. Exponentiation.

In each of the hidden layer basis units, there are 23 additions, 13 multiplications, and 1 exponentiation for a total of 37 operations. Assuming eight basis units per network, the hidden layers perform 296 operations total. The linear summation involves 8 multiplications and 8 additions for each network. The complexity summary is given below. Note that the numbers for the RBF network are doubled since two networks were used: one for *XEL* and one for *EL*.

Table 7: Complexity of two single-output RBF networks pointing-only case

Phase	Additions	Multiplications	Other	Total
Euclidean distances	184	96	0	280
Spread factor mult		8		8
Exponentiation			8	8
Subtotal	184	104	8	296
Matrix multiplication	7	8	0	15
Bias weight addition	1			1
Subtotal	192	112	8	312
x 2 networks	384	224	16	624

Table 8: Complexity of one dual-output RBF network pointing-only case

Phase	Additions	Multiplications	Other	Total
Euclidean distances	184	96	0	280
Spread factor mult		8		8
Exponentiation			8	8
Subtotal	184	104	8	296

Matrix multiplication	14	16	0	30
Bias weight addition	2			2
Totals	200	120	8	328

Table 9: RBF network complexity in the pointing only case

ALGORITHM	ADDITIONS	MULTIPLICATIONS	OTHER	TOTAL
Interpolated least squares	1127	588	0	1715
Two single-output RBF neural networks	384	224	16	624
One dual-output RBF neural network	200	120	8	328

As shown in Chapter 7, RBF networks yield accuracy that is virtually indistinguishable from that of the interpolated least squares algorithm at only about one-fifth to one-third the computational complexity on the pointing-only problem. However, RBF networks exhibit much higher complexity when both pointing and defocus are taken into account, while multilayer perceptron networks maintain their low computational complexity. This is illustrated in Section 6.1.3, where RBF networks grow in complexity when we begin adding defocus to the estimation problem. The growth of RBF network complexity could be attributable to the fact that RBF networks form local approximations while multilayer feedforward networks form global approximations [17].

The disadvantages above do not mitigate the interpolated least-squares algorithm's value as a benchmark against which neural networks may be compared. The disadvantage of slow lookup table performance could be corrected by conversion of the MATLAB code into C and compilation of the resulting C code. Alternately, the entire implementation could be coded directly in a compiled language to yield even faster performance. This algorithm remains a viable contender for implementation on DSN antennas in addition to being a benchmark for comparison. However, the superb performance, good adaptability, and significantly reduced computational complexity of neural networks make them very strong candidates as well.

5.2 Two Types of Neural networks

RBF networks are commonly used to construct highly localized approximations to a function [17], in contrast to the more global approximations constructed by multilayer feedforward networks. Given that neural networks are used only to correct small pointing errors and small errors in subreflector position, the range of input is bounded. Like multilayer feedforward networks, RBF networks offer adaptability to changes in the antenna over time. By contrast, the lookup tables for the interpolated least squares algorithm would need to be re-generated if significant changes were to occur in the antenna over time. This would require an entire new set of raster scans and would result in significant antenna downtime for data collection. Furthermore, the OLS algorithm yields very fast training since the number of iterations of the main algorithm is bounded by the number of samples in the training set [5].

RBF networks are not without disadvantages. If the input vector lies far outside the bounded input range of training, then all of the basis units will have output near zero. Although this does not typically occur as long as the antenna is operating well within the region of interest, this does indicate a weakness in the generalization power of this localized approximation. Furthermore, selection of the basis unit width to be used in all basis units has a crucial impact on performance in real-world experiments, and some degree of human judgment or of brute-force simulation testing is sometimes necessary to find the appropriate range of basis widths.

Multilayer feedforward networks offer computational simplicity and good adaptability compared to the interpolated least squares algorithm. Their chief advantages over RBF networks are twofold. First, performance is significantly less sensitive to network design parameters than that of RBF networks. Although RBF networks exhibit sensitivity to radial basis width selection, multilayer feedforward networks have shown relatively little sensitivity to the number of hidden-layer neurons. This robustness is an important feature since it is often difficult to determine, in the absence of formal criteria, how many hidden-layer units a multilayer perceptron should have or how wide the basis units of an RBF network should be. Additionally, since the multilayer feedforward network does a better job of global generalization than the RBF network does, it is more robust with respect to novel operating conditions.

6 Simulation Results

This chapter focuses on two groups of simulation results. The first group, discussed in Section 0, is a test of the interpolated least squares algorithm. In Chapters 3 and 4, three major assumptions were made. First, it was assumed that the relationship between the aperture plane and focal plane scalar fields can be approximated by a two-dimensional Fourier transform, which is true for ideal paraboloid-hyperboloid antennas [57] but is not strictly true for the antenna under consideration since this antenna is a shaped-reflector system [2,22,23,39]. Second, it was assumed that the horns can be approximated by point sampling. The earlier analysis by Zohar and Vilnrotter was based upon the first two assumptions [57], and these assumptions played a key role in our analysis. Accordingly, a group of simulations aimed at examining the effects of the first two assumptions on algorithm performance is discussed in Section 0, which also tests the quality of quadratic interpolation vis-à-vis other interpolation methods. Third, the effects of subreflector movement along the *z*-axis were modeled entirely by the concept of an equivalent parabolic deformation of the main reflector, but higher order phase changes also exist to a lesser extent. For this reason, real-world experiments provide an important "sanity-check" on subreflector position estimation and control. The simulations described in Section 0 are based on a "test grid" methodology which will be described in that Section.

Section 6.2 is closely related to the real-world experimental results obtained in Chapter 7. The methodology differs from that of Section 0 since we now focus not on RMS accuracy obtained over a test grid of points in *(XEL,EL,P)* space but on RMS accuracy obtained while simulating the acquisition and tracking process. This is a much more relevant performance measure because it directly corresponds to real-world SNR maximization. While the test grid methodology measures the validity of the point sampling and Fourier transform assumptions made in earlier chapters, the simulation of acquisition and tracking operations is meant to provide a measure of how these algorithms would perform in the real-world. Comparison of these results to the results of Chapter 7 yields insight into the accuracy of some of our modeling assumptions and of the overall physical optics simulation model.

In the case of acquisition and tracking simulations, very good RMS tracking performance may be obtained even if performance over a test grid is seemingly poor. This stems from the fact that during acquisition, it is not essential to determine the exact error offset. As long as the *direction* of the error offset is accurately determined and the magnitude of the estimate is not excessively large, it is possible for the algorithm to pull the antenna toward the optimal operating point in a series of steps. Then accuracy near the origin of the test grid proves crucial in obtaining accurate tracking and good SNR maintenance after acquisition. A discussion of this was presented in Section 4.6.

6.1 RMS errors over a test grid

The results of this section focus on estimation of *(XEL,EL,P)* over a test grid of points similar to that shown in Figure 4.1. Due to the architecture of the physical optics simulation software, all focal errors were simulated by adjusting *P*, the parabolic deformation parameter defined in Section 4.3.2. For this reason, although our discussion in Chapters 1 through 4 focused on the ordered tripled *(XEL,EL,Z)*, we will focus on the ordered triple *(XEL,EL,P)* in this chapter. Note that a +1.0-millimeter change in *P* is equivalent to a -2.63-millimeter change in *Z*, the *z*-axis position of the subreflector, for the case of the 70-meter antenna as shown in Section 4.3.2. The parameter *P* is the defocus parameter of interest in all simulation results in the current Chapter.

6.1.1 Characterizing the Interpolated Least squares Algorithm Part I: Testing the Physical Assumptions

Simulations in this group were run under three broad sets of conditions. In set 1, a simulated focal plane field was generated in the following manner. First, actual dish deformation data obtained via holography [22,37,40-42] were taken as being samples of the deformation function $d\left(r,\theta\right)$. Equation (0.0.34) was then applied to obtain the effective aperture field $U_a\left(r,\theta\right)$. The resulting aperture field was transformed, using a two-dimensional FFT with appropriate scaling, to obtain the focal plane field matrix. The focal plane field matrix was sampled at seven points corresponding to the seven horn centers based on the pointing offset being simulated to obtain AFCS voltage vectors. Both reference tables for the interpolated least squares algorithm and simulated input vectors to test estimation accuracy were generated in this manner. In set 2, the manner of sampling the focal plane field was changed from the point sampling method of set 1 to realistic horn sampling. A comparison of these two sets yields one method of testing the point sampling assumption made in Chapter 3. In set 3, the manner of focal plane field generation was changed from Fourier transforms to physical optics simulations. The comparison between Fourier-transform based results and physical optics simulation based results helps in judging the quality of the Fourier-transform based model as it relates to the given estimation problem.

The reference table was a three-dimensional table similar to that illustrated in Figure 4.1. Data points were taken at offsets spaced 1-millidegree apart in both *XEL* and *EL* ranging from -3.0-millidegrees to +3.0-millidegrees in both pointing directions. Sampling was carried out at 1.0-millimeter intervals in *P* with parabolic deformation coordinate *P* ranging from -5.0-millimeters to +5.0-millimeters. This is the equivalent of having the subreflector range from Z = +13.15-millimeters to Z = -13.15 millimeters in -2.63-millimeter increments. The test pattern, or test grid, is the set of points in *(XEL,EL,P)* spaced used for testing purposes. In this case, it was also a three dimensional rectangular grid whose form is similar to that of Figure 4.1. Test offsets ranged from -1.50- to +1.50-millidegrees in both *XEL* and *EL* in steps of 0.50-millidegrees. The parabolic deformation parameter *P* ranged from -2.5-millimeters to

+2.5-millimeters in 0.5-millimeter steps, which means that the equivalent subreflector position *Z* ranged from +6.575-millimeters to -6.575-millimeters in -1.315-millimeter steps. We note that five of the parabolic deformation parameters: -2.0, -1.0, 0.0, 1.0, and 2.0-millimeters, were defocus parameters previously seen in the reference set. These are referred to as previously seen defocus parameters. The six defocus parameters: -2.5, -1.5, -0.5, 0.5, 1.5, and 2.5-millimeters, were defocus parameters that were not included in the reference table, and these are referred to as previously unseen defocus parameters.

Although quadratic interpolation was described in Section 4.4, cubic spline interpolation was also used in order to determine whether better interpolation accuracy could have been achieved. Results labeled with "cubic" refer to those obtained with the interpolated least squares algorithm of Section 4.4 using cubic spline interpolation. Results labeled with "quad" refer to those obtained with the quadratic interpolation method of that section. Furthermore, the label "FP" means "Fourier point". It refers to Fourier-transform generated focal plane fields with point sampling, or set 1 data. The label "FH" means "Fourier horn". It refers to Fourier-transform generated focal plane fields with realistic horn sampling, or set 2 data. The label "PO" means "physical optics". It refers to focal plane fields generated using a physical optics simulation, or set 3 data.

RMS errors in pointing are illustrated from Figure 6.1 through Figure 6.3. RMS errors in estimating parabolic deformation, or *P*, are illustrated from Figure 6.4 through Figure 6.6. These figures illustrate several interesting points. First, cubic spline interpolation yields accuracy similar to that of simpler quadratic interpolation in estimating pointing errors, while quadratic interpolation appears to be more robust in the problem of defocus estimation (for the Fourier transform cases). Second, in the problem of defocus estimation, RMS errors were lower for the most realistic case, corresponding to a physical optics simulation with horn sampling, than they were for the two Fourier transform cases even though the same parameters were applied to all three simulation data sets. Third, RMS pointing estimation errors are slightly higher for the physical optics case than for the two Fourier transform cases. While the results of this section would seemingly suggest that simple quadratic interpolation is slightly better than cubic spline interpolation in certain cases, cubic spline interpolation achieves its best accuracy near the origin of *(XEL,EL,P)*-space due to the presence of many neighboring points. By contrast, cubic-spline interpolation also yields its weakest estimation performance as we approach the edges of the reference table in *(XEL,EL,P)*-space. Quadratic interpolation will yield similar performance throughout most of *(XEL,EL,P)*-space since it always relies only on immediate neighbors. This issue is mentioned again in Section 6.2 where cubic-spline interpolation yields better performance near the origin of *(XEL,EL,P)*-space than does quadratic interpolation.

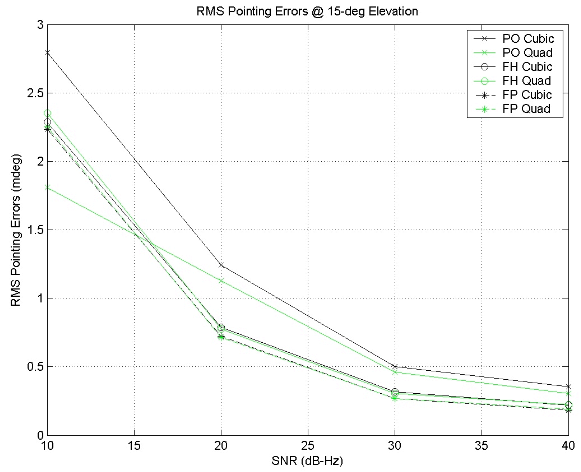

Figure 6.1: RMS Pointing Errors in Millidegrees at 15-degrees Elevation

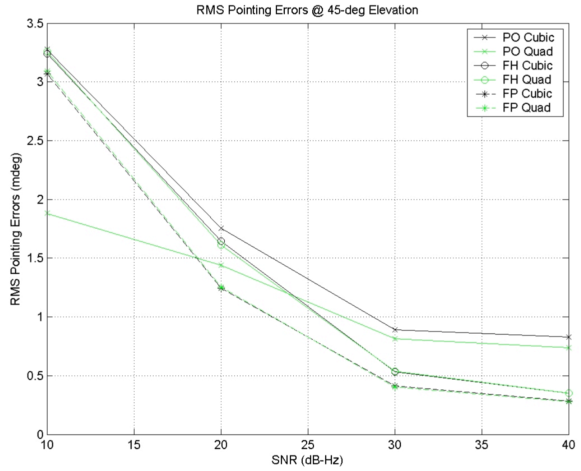

Figure 6.2: RMS Pointing Errors in Millidegrees at 45-degrees Elevation

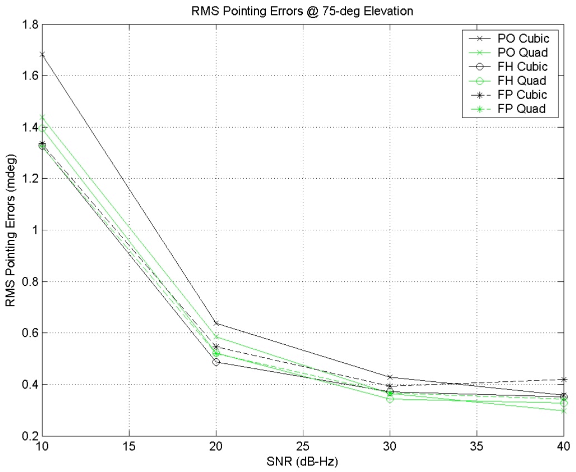

Figure 6.3: RMS Pointing Errors in Millidegrees at 75-degrees Elevation

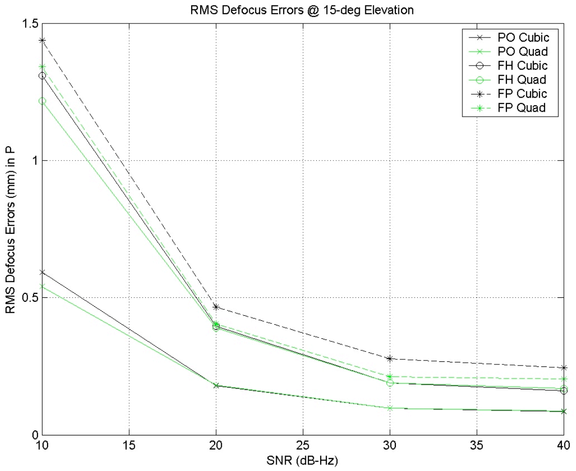

Figure 6.4: RMS Defocus Errors *(P)* in Millimeters at 15-degrees Elevation

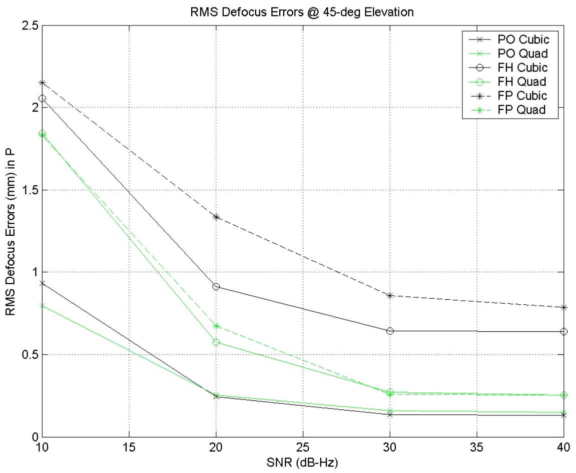

Figure 6.5: RMS Defocus Errors *(P)* in Millimeters at 45-degrees Elevation

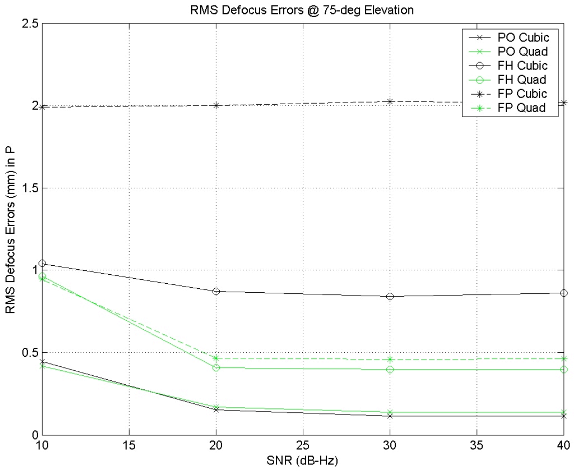

Figure 6.6: RMS Defocus Errors *(P)* in Millimeters at 75-degrees Elevation

Results at 40 dB-Hz are summarized in Table 10 and in Table 11. From these tables, we note that pointing accuracy typically exceeds the DSN's 0.8-millidegree requirement in all cases but one: 45 degrees elevation using a physical optics simulation. Here, pointing accuracy lies within the 0.8-millidegree requirement with quadratic interpolation but is worse than it was for the Fourier-transform cases.

The differences in defocus estimation accuracy indicate that the interpolated least-squares algorithm is more robust in the realistic physical optics simulations than it was in the Fourier-transform based simulations. This algorithm is, however, less robust in the physical optics pointing case, although its performance with physical optics simulations is close to its performance with Fourier-transform based simulations at 15 degrees and at 75 degrees. Note that we provide focal errors in both *P* (parabolic deformation coordinate) and in *Z* (subreflector offset coordinate). The two are related since a 1.0-millimeter movement in *P* is equivalent to a 2.63-millimeter movement in *Z* in the opposite direction. We have provided both since much of our discussion in Chapters 1 through 4 focused on the *Z*-coordinate, which is a controllable parameter (unlike *P* which refers to main reflector deformation).

Table 10: RMS Pointing Errors in Millidegrees at 40 dB-Hz

Antenna Elevation (deg)	PO Cubic	PO Quad	FH Cubic	FH Quad	FP Cubic	FP Quad
15	0.3527	0.3065	0.2204	0.2229	0.1832	0.1903
45	0.8264	0.7386	0.3488	0.3515	0.2824	0.2804
75	0.3566	0.3000	0.3526	0.3278	0.4203	0.3421

Table 11: RMS Defocus Errors *(P)* in Millimeters at 40 dB-Hz

Antenna Elevation (deg)	PO Cubic	PO Quad	FH Cubic	FH Quad	FP Cubic	FP Quad
15	0.0840	0.0867	0.1609	0.1685	0.2441	0.2036
45	0.1313	0.1500	0.6396	0.2555	0.7878	0.2525
75	0.1142	0.1376	0.8598	0.3971	2.0177	0.4632

Table 12: Equivalent RMS Subreflector Errors *(Z)* in Millimeters at 40 dB-Hz

Antenna Elevation (deg)	PO Cubic	PO Quad	FH Cubic	FH Quad	FP Cubic	FP Quad
15	0.2209	0.2280	0.4232	0.4432	0.6420	0.5355
45	0.3453	0.3945	1.6821	0.6720	2.0719	0.6641
75	0.3003	0.3619	2.2613	1.0444	5.3066	1.2182

In summary, the underlying physical assumptions: point sampling and a Fourier transform approximation, are good but not perfect modeling assumptions. Estimation of focal errors can be performed with *greater* accuracy in the most realistic case involving physical optics and horn sampling. Even if pointing performance at previously unknown defocus is relatively poor, good estimation of focal errors often makes it possible to return *P* to a point close to 0.0-millimeters. The pointing correction problem is easier to solve once this is achieved.

Actual analysis of a shaped-reflector system very often involves running physical optics simulations or performing extensive physical optics analysis (just a few examples are in [2,23,37]), and the Fourier transform approximation used by Zohar and Vilnrotter [57] as well as in this work provides a more tractable but less accurate analytical model. Nevertheless, simulation results for Fourier transform models are similar to those for physical optics models in the majority of cases except for the 75-degree defocus estimation case (and, to a lesser extent, the 45-degree defocus estimation case although quadratic interpolation still yielded good performance even here). The similar performance of the interpolated least squares algorithm under Fourier transform and physical optics models suggests that the

Fourier transform model is still a good model for pointing analysis, but its behavior under severe deformation conditions at 75-degrees is a topic of future research.

6.1.2 Interpolation Issues

One possible criticism of our interpolated least squares estimator is that quadratic interpolation is suboptimal. It could be argued that better interpolation methods would yield better results. In this section, we compare three interpolation methods: sinc-function interpolation, cubic-spline interpolation, and quadratic interpolation.

The following tables present pointing errors and defocus estimation errors at 15, 45, and 75 degrees elevation with an SNR of 40 dB-Hz for the following three interpolation algorithms: sinc-function interpolation, cubic-spline interpolation, and quadratic interpolation.

Table 13: Pointing estimation errors in millidegrees over all defocus levels

Algorithm	15 degrees	45 degrees	75 degrees
Sinc-interpolation	0.6527	0.8348	0.6502
Cubic-spline interpolation	0.3527	0.8264	0.3566
Quadratic interpolation	0.3065	0.7386	0.3000

Table 14: Pointing estimation errors in millidegrees with previously seen defocus

Algorithm	15 degrees	45 degrees	75 degrees
Sinc-interpolation	0.6072	0.6281	0.6014
Cubic-spline interpolation	0.2290	0.3172	0.2875
Quadratic interpolation	0.1633	0.2570	0.2177

Table 15: Pointing estimation errors in millidegrees with previously unseen defocus

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.6887	0.9748	0.6886
Cubic-spline interpolation	0.4298	1.0816	0.4057
Quadratic interpolation	0.3876	0.9729	0.3546

From Table 14 it is clear that for previously seen defocus levels pointing accuracy lies very much within the DSN 0.8-millidegree requirement. For previously unseen defocus, however, Table 15 shows us that the pointing accuracy requirement is not met at 45 degrees although it is certainly met at 15 degrees and at 75 degrees. Referring back to Table 10 where the DSN's 0.8-millidegree requirement was violated at 45 degrees with cubic spline interpolation, we now understand the problem: the interpolated least squares estimator does not always perform well with previously unseen defocus values.

Table 16: Defocus estimation errors *(P)* in millimeters over all defocus levels

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.2052	0.2015	0.1734
Cubic-spline interpolation	0.0840	0.1313	0.1142
Quadratic interpolation	0.0867	0.1500	0.1376

Table 17: Defocus estimation errors *(P)* in millimeters with previously seen defocus

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.1927	0.1905	0.1739
Cubic-spline interpolation	0.0147	0.0209	0.0232
Quadratic interpolation	0.0890	0.1249	0.1264

Table 18: Defocus estimation errors *(P)* in millimeters with previously unseen defocus

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.2152	0.2104	0.1730
Cubic-spline interpolation	0.1130	0.1768	0.1532
Quadratic interpolation	0.0848	0.1682	0.1463

Table 19: Equivalent subreflector errors *(Z)* in millimeters over all defocus levels

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.5397	0.5299	0.4560
Cubic-spline interpolation	0.2209	0.3453	0.3003
Quadratic interpolation	0.2280	0.3945	0.3619

Table 20: Equivalent subreflector errors *(Z)* in millimeters with previously seen defocus

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.5068	0.5010	0.4574
Cubic-spline interpolation	0.0387	0.0550	0.0610
Quadratic interpolation	0.2341	0.3285	0.3324

Table 21: Equivalent subreflector errors *(Z)* in millimeters with previously unseen defocus

Algorithm	**15 degrees**	**45 degrees**	**75 degrees**
Sinc-interpolation	0.5660	0.5534	0.4550
Cubic-spline interpolation	0.2972	0.4650	0.4029
Quadratic interpolation	0.2230	0.4424	0.3848

In the pointing case the quadratic interpolated algorithm's performance is almost identical to that of the cubic spline interpolated version, but the sinc-function interpolated version yields the worst performance. The main problem observed in the pointing and defocus cases is this: there is a systematic bias in the sinc-interpolated version of the algorithm. The sinc-function case deserves further examination as it is very important: for an infinite lookup table with Nyquist spacing, sinc-function interpolation is optimal [4,56].

Consider the use of sinc-function interpolation in the pointing-only case without defocus. If a modified table, which measures from -8.0- to +8.0-millidegrees in both XEL and EL in 1.0-millimeter steps, is used, we obtain the "extended table" results. The "standard table" results are obtained using a table from -3.0- to +3.0-millidegrees in both XEL and EL. A comparison is shown in the table below.

Table 22: Comparison of pointing estimation quality in the no-defocus case with two lookup table sizes at 45 degrees using sinc-function interpolation

Table size	XEL (mean/std)	EL (mean/std)	Total (rms)
Extended (17x17)	0.4380/0.1267	0.4472/0.1349	0.6527
Standard (7x7)	0.4015/0.2007	0.4004/0.2054	0.6356

The use of the extended lookup table results in a reduction in random error (standard deviation) as expected, but there is always a systematic error of over 0.4-millidegrees. Although RMS error performance lies within DSN requirements, we note that it can be improved by calculating and subtracting out the systematic (mean) error. In that case, the sinc-function interpolation method would yield errors comparable to those of the other two algorithms. The exact cause of this error is not fully understood, however, and we regard this as a topic for future research. We note, from Section 4.2, that the function $g_{\theta_{EL}}(XEL, EL, Z)$ defined by equation (0.0.64) has unknown frequency content. This is one reason why sinc-function interpolation may not yield better performance than the cubic-spline and quadratic interpolation methods. The true Nyquist interval may be smaller, or there may be tail effects that are not accounted for in the use of sinc-functions since sinc-function based reconstruction is based on the ideal case of an infinite lookup table. The reduction in *random* pointing error with a larger lookup table in Table 22 suggests that lookup table size must be increased to compensate for this “tail effect.

A practical limitation prevents the gathering of very large lookup tables. If the table is made too long along the *XEL* and *EL* dimensions, we will reach a point where much of the power of the focal plane field will no longer be in the horns. At this point, received SNR will be low, and there will be considerable errors in any lookup table points at the edges. This will cause degradation of lookup table accuracy and a loss of performance. Sinc-function interpolation will not be considered further in this dissertation.

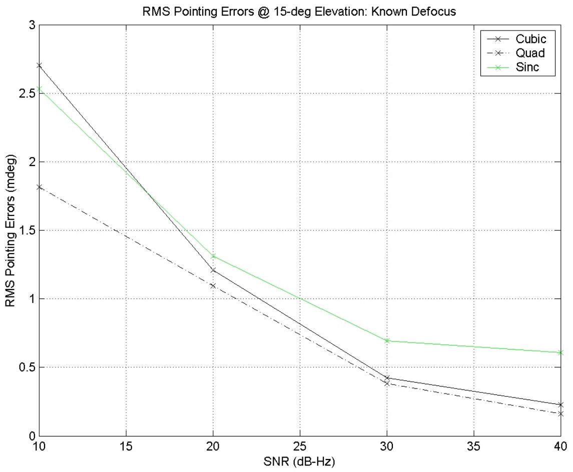

Figure 6.7: RMS Pointing Errors with Previously Seen Defocus at 15-degrees Elevation

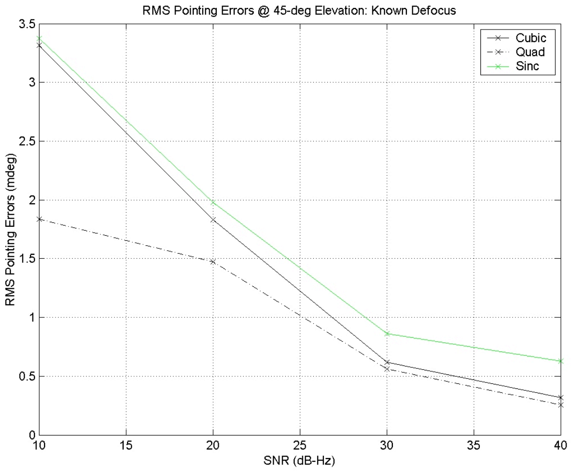

Figure 6.8: RMS Pointing Errors with Previously Seen Defocus at 45-degrees Elevation

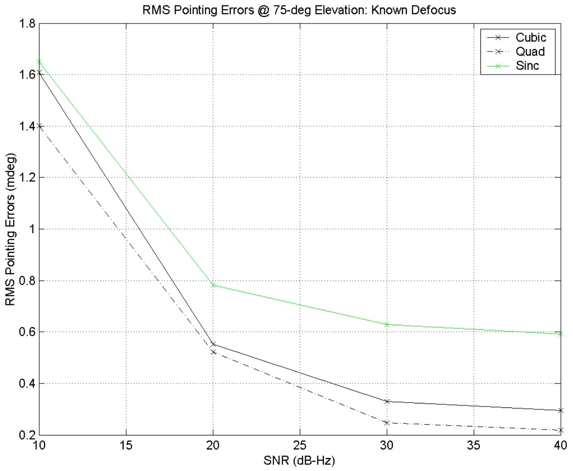

Figure 6.9: RMS Pointing Errors with Previously Seen Defocus at 75-degrees Elevation

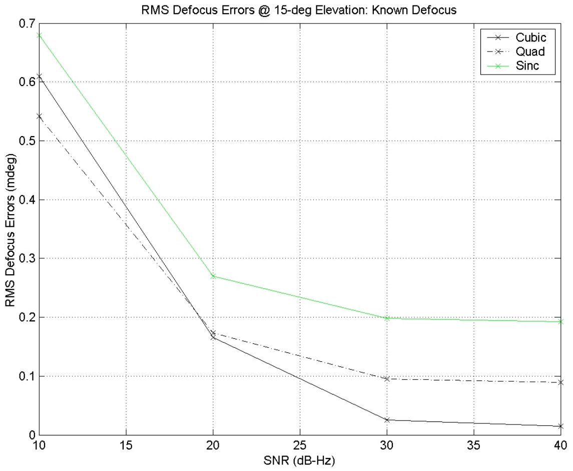

Figure 6.10: RMS Defocus Estimation *(P)* Errors with Previously Seen Defocus at 15-degrees Elevation

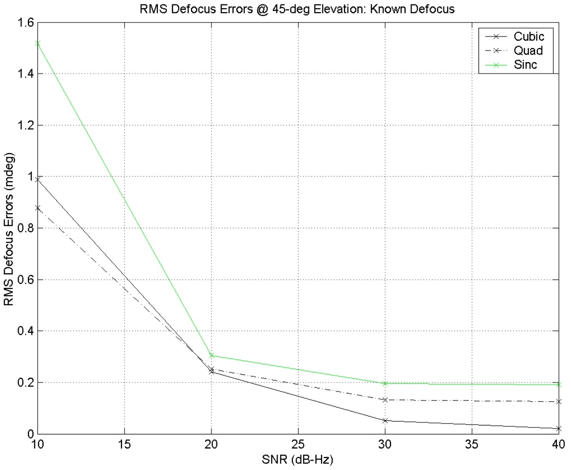

Figure 6.11: RMS Defocus Estimation Errors *(P)* with Previously Seen Defocus at 45-degrees Elevation

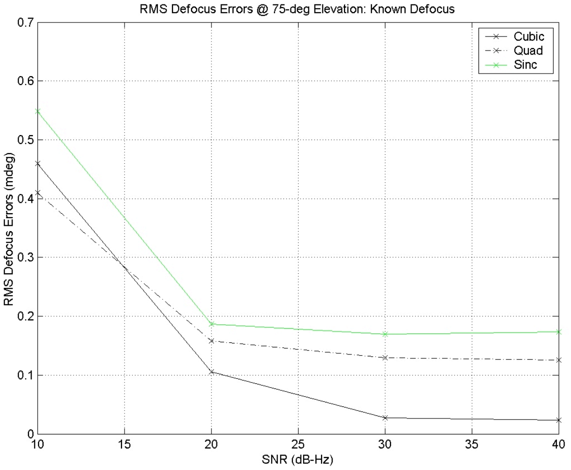

Figure 6.12: RMS Defocus Estimation Errors *(P)* with Previously Seen Defocus at 75-degrees Elevation

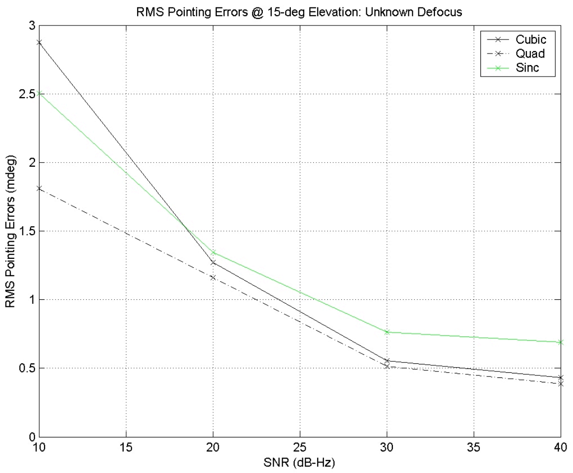

Figure 6.13: RMS Pointing Errors with Previously Unseen Defocus at 15-degrees Elevation

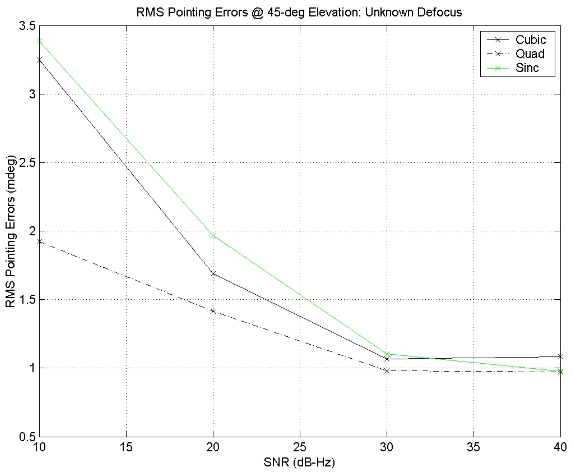

Figure 6.14: RMS Pointing Errors with Previously Unseen Defocus at 45-degrees Elevation

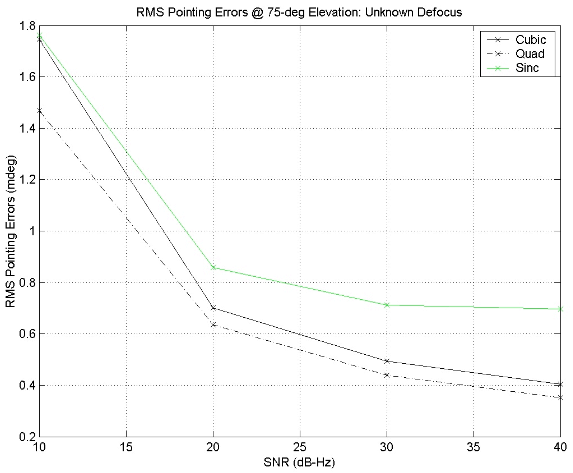

Figure 6.15: RMS Pointing Errors with Previously Unseen Defocus at 75-degrees Elevation

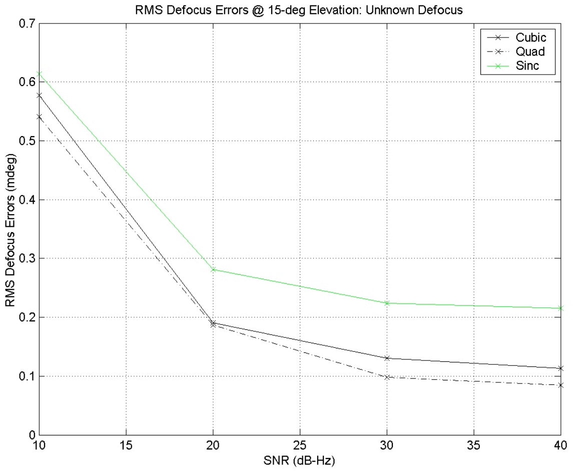

Figure 6.16: RMS Defocus Estimation Errors *(P)* with Previously Unseen Defocus at 15-degrees Elevation

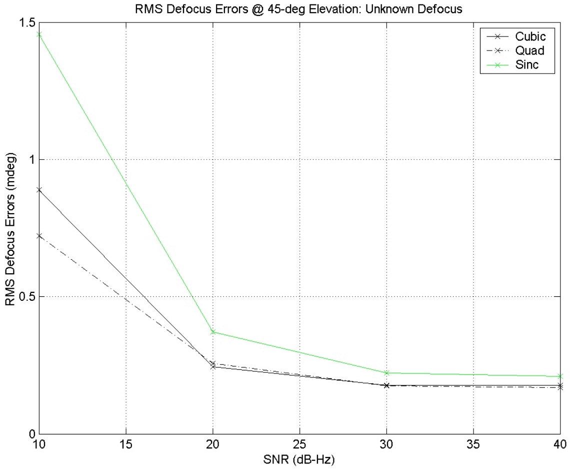

Figure 6.17: RMS Defocus Estimation Errors *(P)* with Previously Unseen Defocus at 45-degrees Elevation

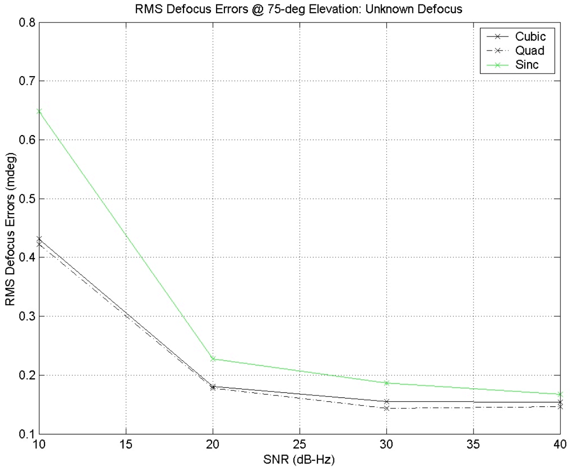

Figure 6.18: RMS Defocus Estimation Errors *(P)* with Previously Unseen Defocus at 75-degrees Elevation

In summary, the quadratic interpolated least squares algorithm yields estimation accuracy over a test grid comparable to that of the cubic-spline interpolated version and superior to that of the sinc-interpolated version. This algorithm was used as the benchmark against which neural networks were compared during the course of experiments detailed in Chapter 7. The cubic-spline interpolated least squares algorithm is included in acquisition and tracking simulations (Section 6.2) but, due to time constraints, was not tested in real-world experiments (Chapter 7).

6.1.3 A comparison between the quadratic interpolated least squares algorithm and neural networks

A set of neural networks had been trained using reference tables based on physical optics simulations with horn sampling. Both multilayer feedforward and RBF neural networks were trained under these conditions, and the results are presented below. We note that these results were taken using neural networks that had not been adapted in response to data gathered during real-time operation, and adaptive tracking results are deferred until Section 6.2. For this reason, the results presented here do not illustrate a primary advantage of neural networks: adaptability. Nevertheless, these results do provide an indication of how a neural network's performance compares to that of the interpolated least squares algorithms when neural networks are trained using the same data used in the interpolated least squares lookup tables.

In all simulations discussed in this Chapter, the following neural network designs were used. Multilayer feedforward networks were designed with 30 hidden layer neurons and three outputs for *XEL*, *EL,* and *P*. RBF networks were generated using the OLS algorithm [5]. All RBF networks had three outputs, as did the multilayer feedforward networks. We note that while RBF networks yielded relative computational simplicity in the pointing-only case (see Section 5.1 and [31]), complexity rises when defocus estimation is added to the problem, possibly due to the fact that these networks generate highly localized approximations to the desired pointing and defocus estimator [17].

Table 23: RBF Network Design Parameters

Elevation (deg)	Number of Basis Units	Basis Unit Spread
15	184	1.0
45	258	1.0
75	255	1.0

Although a brief complexity analysis had been given in Section 5.1, we believe it is useful to present actual running times in MATLAB. These running times may not be representative of actual performance with compiled language code, but they do provide a rough indication of one key neural network advantage: low computational complexity.

Before we give running times, we state the conditions under which the simulation was run.

1. The running times are the total CPU time spent computing estimates over the entire test grid, not just at one point of the test grid.
2. The simulated antenna pointing elevation was 75 degrees.
3. The table lookup for the interpolated least squares algorithms was performed just *once*. After this lookup was performed, each of the three interpolation methods was run, yielding three sets of results.
4. In addition, both a multilayer feedforward network and a RBF network were run.

5. The first running time given for the table lookup is required for all three versions of the interpolated least squares algorithm regardless of interpolation method. In all three cases, it is necessary to compare the input vector to each of the reference vectors.
6. So the interpolation running time is *in addition to* the table lookup running time.

The run times are presented below. Here, the terms “sinc”, “cubic”, and “quad” refer to the three versions of the interpolated least squares algorithm discussed previously. The terms “FF” and “RBF” refers to multilayer feedforward networks and to RBF networks, respectively. All times presented in Table 24 are total simulation times over the *entire* test grid. From this table, we see that the table lookup step is the step that causes the greatest performance loss for two of the three interpolated least squares algorithms. All times were found using the MATLAB profiler in MATLAB 6.5.

Table 24: Running times of estimation algorithms

Algorithm	**Table Lookup Time (sec)**	**Interpolation Time (sec)**	**Total Time (sec)**
Sinc	145.308	1313.957	1459.265
Cubic	145.308	33.342	178.65
Quad	145.308	14.905	160.213
FF	N/A	N/A	29.210
RBF	N/A	N/A	32.459

An unexpected result is the fast running time of the RBF network. This network contains 255 hidden layer units, implying a total of 255 Euclidean distance calculations in addition to 255 exponentiations. One expects this complexity to be about one-half of the complexity of 539 Euclidean distance calculations corresponding to the lookup table's 539 entries. There is a possibility this is an implementation-dependent issue, and readers are cautioned that alternate implementations of the Euclidean distance calculations in the table may yield better interpolated least squares performance. However, this implementation had been heavily optimized using the MATLAB profiler to eliminate areas of inefficiency. Nevertheless, interpolation times rival the total running times of neural networks for both sinc-interpolated and cubic-spline interpolated versions of the interpolated least squares algorithm, with only quadratic interpolation time being faster than neural network total running times.

Even more interesting is the fact that the RBF network, despite having a much larger number of hidden layer units than the multilayer feedforward network, has a running time that is only 11.1% longer than that of the multilayer feedforward network. Based upon calculations of the type performed in Chapter 5, one expects the RBF network to have a much longer running time given the large number of hidden layer units. Each of the 30 hidden layer units in the multilayer feedforward network needs to compute both a vector dot product and a hyperbolic tangent, so the complexity of each hidden layer unit of a multilayer feedforward network is expected to be on par with that of a radial basis unit. We believe this effect is implementation dependent, and readers are cautioned that MATLAB-based performance profiles may not reflect performance achieved in the real-world using compiled code or highly optimized assembly language implementations. The results still agree qualitatively with the claim made in Chapter 5 that neural networks offers notably lower running times in real-world implementations.

The neural networks' accuracy at 40 dB-Hz is compared to that of the quadratic interpolated least squares algorithm in both Table 25 and Table 26. At this high SNR, neural networks typically outperform the quadratic interpolated least squares algorithm in most cases except for pointing at 45 degrees antenna elevation. However, even in this case, multilayer feedforward networks still outperform interpolated least squares algorithms although RBF networks do not.

Table 25: Comparison of RMS Pointing Errors in Millidegrees at 40 dB-Hz

Elevation (deg)	**Quadratic Interpolated Least squares (mdeg)**	**Multilayer Feedforward Neural Network (mdeg)**	**RBF Neural Network (mdeg)**
15	0.3065	0.2206	0.1974
45	0.7386	0.5431	1.0765
75	0.3000	0.1731	0.1798

Table 26: Comparison of RMS Defocus Estimation Errors *(P)* in Millimeters at 40 dB-Hz

Elevation (deg)	**Quadratic Interpolated Least squares (mm)**	**Multilayer Feedforward Neural Network (mm)**	**RBF Neural Network (mm)**
15	0.0867	0.0522	0.0274
45	0.1500	0.1536	0.0626
75	0.1376	0.0941	0.1112

Table 27: Comparison of RMS Equivalent Subreflector Position Errors *(Z)* in Millimeters at 40 dB-Hz

Elevation (deg)	Quadratic Interpolated Least squares (mm)	Multilayer Feedforward Neural Network (mm)	RBF Neural Network (mm)
15	0.2280	0.1373	0.0721
45	0.3945	0.4040	0.1646
75	0.3619	0.2475	0.2925

Although the above tables show that neural networks typically have an advantage over interpolated least squares algorithms at high SNR in terms of estimation accuracy, it is also true that the interpolated least squares algorithms yield an advantage in performance at lower SNR which is illustrated from Figure 6.19 through Figure 6.24.

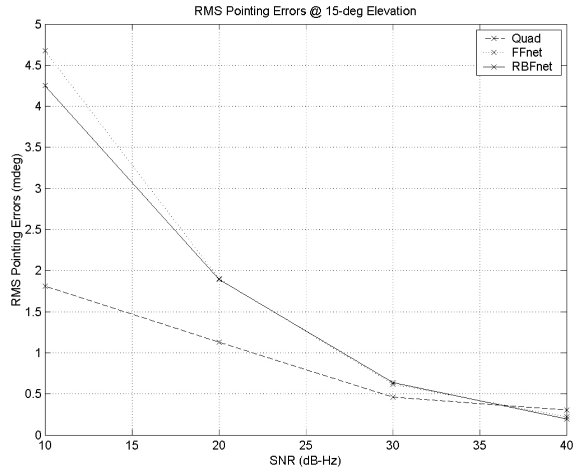

Figure 6.19: RMS Pointing Errors at 15-degrees Elevation

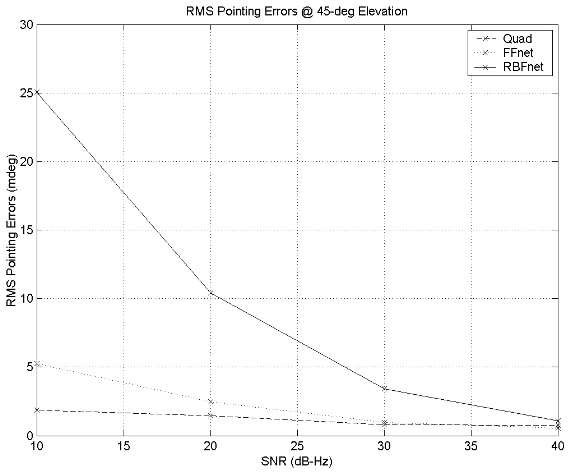

Figure 6.20: RMS Pointing Errors at 45-degrees Elevation

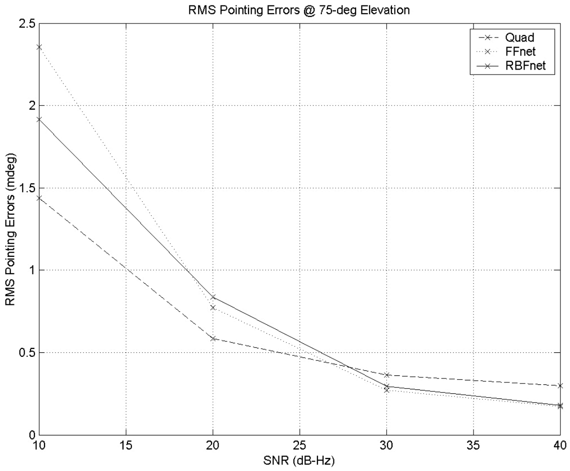

Figure 6.21: RMS Pointing Errors at 75-degrees Elevation

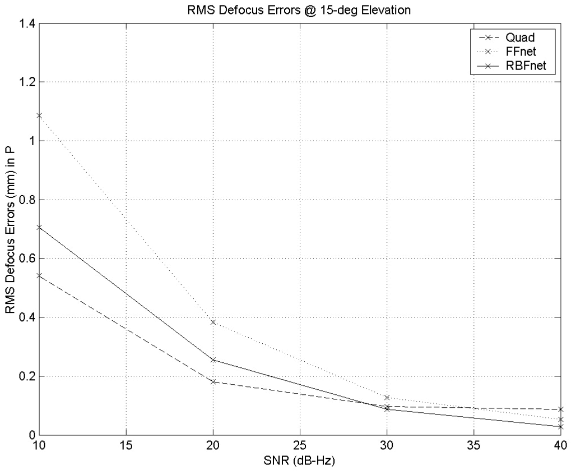

Figure 6.22: RMS Defocus Estimation *(P)* Errors at 15-degrees Elevation

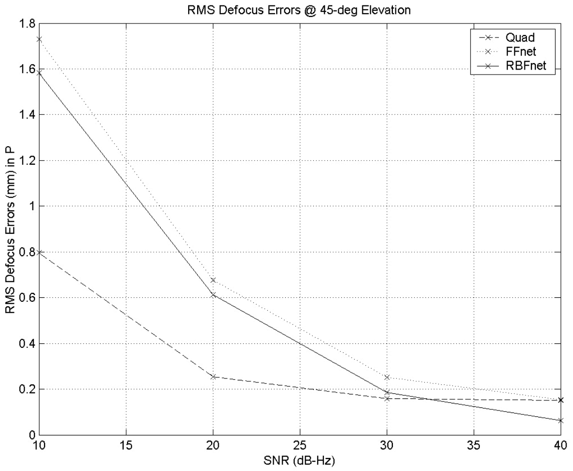

Figure 6.23: RMS Defocus Estimation *(P)* Errors at 45-degrees Elevation

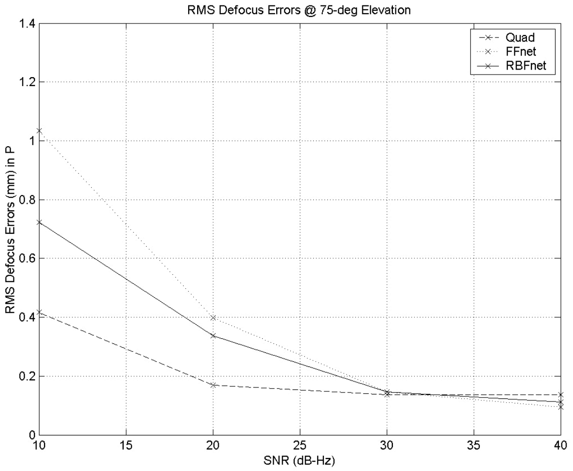

Figure 6.24: RMS Defocus Estimation *(P)* Errors at 75-degrees Elevation

The figures above clearly show that the interpolated least squares algorithm often has an advantage under noisier operating conditions. We also note that the Universal Approximation Theorem states that multilayer feedforward networks can approximate any smooth, continuous, bounded function over a bounded domain with arbitrarily high accuracy. This implies that there exists a multilayer feedforward network of unknown design (the Universal Approximation Theorem is strictly an existence theorem) that can achieve better estimation accuracy than that of the interpolated least squares algorithm, and this is reflected in the superior high SNR performance of the multilayer feedforward network.

6.2 Acquisition and Tracking Simulations

Acquisition is relatively robust with respect to estimation errors away from the origin, but tracking is sensitive to errors in estimation near the origin. This set of simulations, based upon the experimental methods used in obtaining the real-world results of Chapter 7, is based on simulated acquisition and tracking operations. While the results of Section 0 are important if one uses a "single-shot" method of correcting pointing and focus, the discussion of Section 4.6 gives us reasons to apply only partial corrections, effectively computing a weighted time average that is more robust with respect to estimation errors. All acquisition and tracking simulation results presented in this section are based on focal plane fields generated using physical optics simulation code, and all sampling of the focal plane field is performed using horn functions as opposed to point sampling. Both cubic-spline and quadratic interpolated versions of the interpolated least squares algorithm were tested alongside RBF and multilayer feedforward neural networks.

In each simulation, an offset was applied to the antenna in *(XEL,EL,P)*-space. A simple example of this procedure was illustrated in Section 4.6. On each iteration, the following steps were performed:

1. An algorithm (interpolated least squares or neural network) was used to estimate the offset.
2. The estimate was multiplied by a *pointing gain* variable α , which was 0.7 in this case. The result was then subtracted from the antenna's current absolute offset.
3. Either return to step 1 or stop if the target number of iterations has been reached.

Eight initial offsets were used at each of three antenna pointing elevations. Each of the following offsets is in units of millidegrees for pointing and millimeters for parabolic deformation P. Note that the 2.5-millimeter offsets in P translate to 6.58-millimeter offsets in subreflector position coordinate Z.

1. (-3,-3,-2.5)
2. (-3,-3,+2.5)
3. (-3,+3,-2.5)
4. (-3,+3,+2.5)
5. (+3,-3,-2.5)
6. (+3,-3,+2.5)
7. (+3,+3,-2.5)
8. (+3,+3,+2.5)

Here in Section 6.2 only, we did *not* make (0,0,0) the point of maximum SNR as is done in all other parts of this dissertation. Instead, the reference tables had their (0,0,0) point slightly away from the true maximum, and all simulations were normalized so that (0,0,0) would be the 40 dB-Hz point. Hence, the interpolated least squares algorithms would tend to track toward this point even though it is slightly suboptimal. We wish to examine the behavior of both neural networks and the interpolated least squares algorithm under realistic conditions, and we also wish to illustrate the main advantage of neural networks: adaptability. The ability to learn from real-time experience without having to gather an entire new raster scan is a key advantage of neural networks. The retraining procedure used for neural networks is as follows:

1. The neural network is permitted to perform acquisition and tracking operations starting from each of the eight offsets listed above. In each case, 30 data points are gathered for a total of 240 data points.
2. The 240 data points are organized according to SNR achieved at each point, and the point with the highest SNR is designated as the new (0,0,0) point. Then all (XEL, EL, P) tracking data are re-centered relative to this point.
3. The network undergoes 100 epochs of backpropagation training using the real-time training set. Here, an "epoch" is defined to be the presentation of each of the 240 training points, in random order, to the network exactly once during the backpropagation process. In the case of RBF networks, only the linear combiner is retrained (using LMS) [13,16-18,31].
4. The retrained network is again allowed to acquire and track from the same eight offsets, and the procedure may be repeated as often as desired

Note that although many iterations may be used in retraining, these can be multitasked alongside real-time tracking. By creating an identical copy of the tracking network for training purposes, we can be retraining one copy while performing acquisition and tracking operations with the other. This is more efficient than gathering more raster scan data, which is a time-consuming process. Furthermore, tracking with a neural network while retraining another copy of that network is not a difficult task for most modern computers.

6.2.1 Example at 75-degrees Antenna Elevation

This example highlights the learning capabilities of neural networks. We recall that at 75 degrees, severe main reflector deformation causes major focal plane field distortion as shown in Figure 1.6. As a result, the (0,0,0) point, which corresponds to the point of maximum received SNR, will be shifted in pointing and defocus space. Although we used the original (0,0,0) point for the undistorted case as the reference point in our interpolated least squares reference set, which was also used as the training set for the neural networks, this is not the point of maximum SNR. The challenge here is this: the neural network must learn the true (0,0,0) point by retraining with data gathered during real-time acquisition and tracking operations.

For comparison purposes, we first present the performance of the interpolated least squares algorithm using cubic spline interpolation. The SNR at (0,0,0) was 40 dB-Hz, and the starting offset was (3,3,2.5) in (XEL, EL, P)-space (in millidegrees and millimeters). The performance of the interpolated least squares algorithm using cubic spline interpolation is shown from Figure 6.25 through Figure 6.27. The performance of a multilayer feedforward network prior to retraining is shown from Figure 6.28 through Figure 6.30. The performance of the same multilayer feedforward network after retraining is illustrated from Figure 6.31 through Figure 6.33. In all cases, 30 iterations were used for acquisition and tracking. The first 10 iterations are considered to be the acquisition stage, while the last 20 iterations are regarded as the tracking stage.

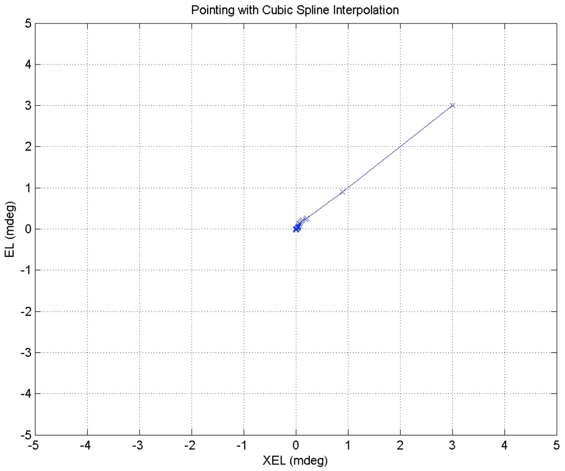

Figure 6.25: Acquisition and Tracking in Pointing Space with Cubic Spline Interpolated Least squares Algorithm

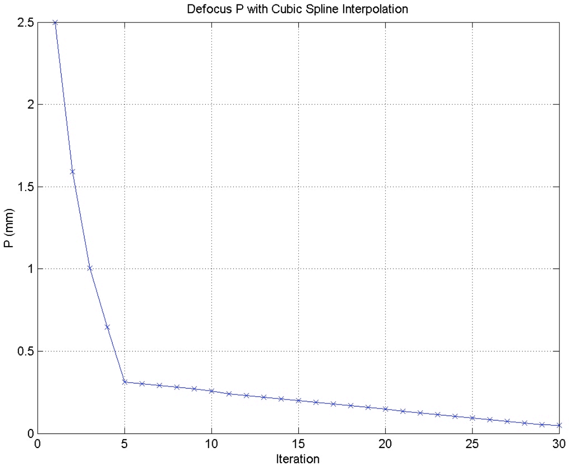

Figure 6.26: Defocus *P* in millimeters

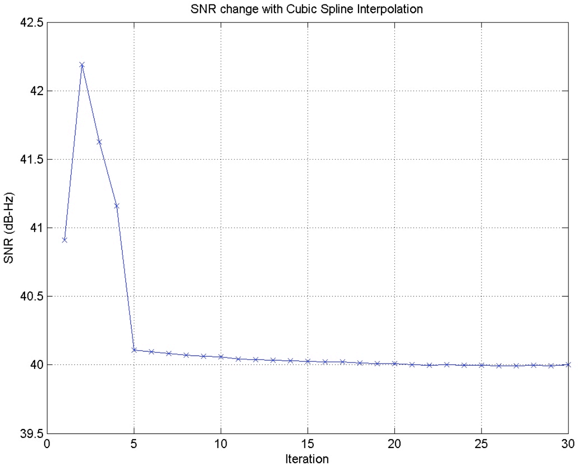

Figure 6.27: SNR as a Function of Time with Cubic Spline Interpolated Least squares Algorithm

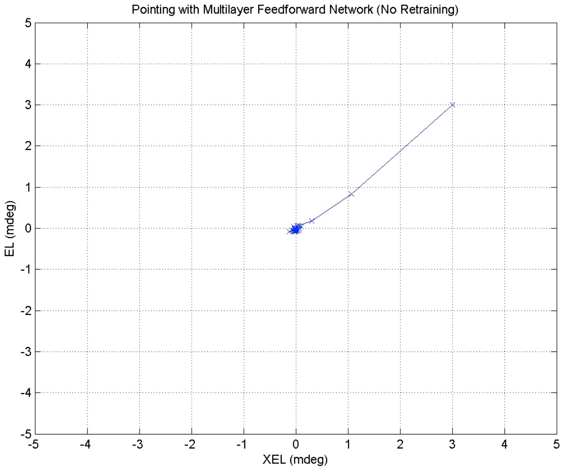

Figure 6.28: Acquisition and Tracking in Pointing Space with a Multilayer Feedforward Network Prior to Retraining

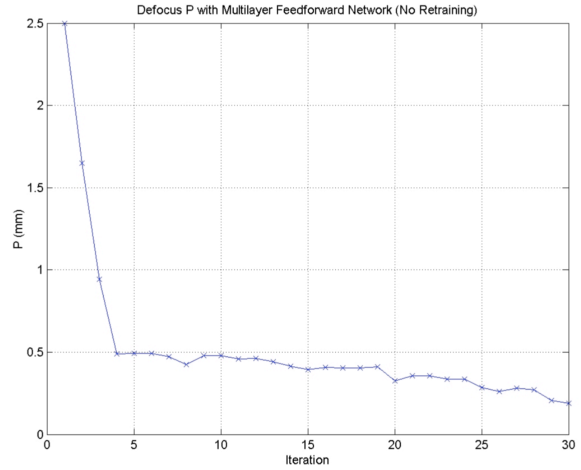

Figure 6.29: Defocus Correction with a Multilayer Feedforward Network Prior to Retraining

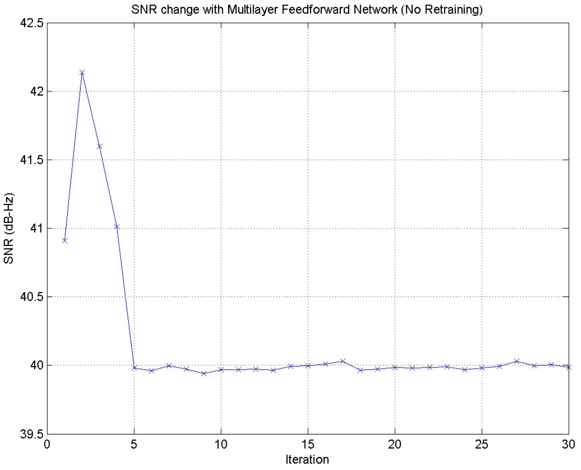

Figure 6.30: SNR as a Function of Time with a Multilayer Feedforward Network Prior to Retraining

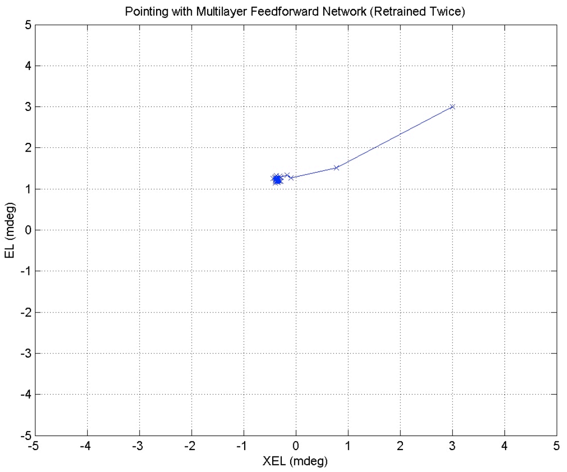

Figure 6.31: Acquisition and Tracking in Pointing Space with a Multilayer Feedforward Network after Two Retraining Sessions

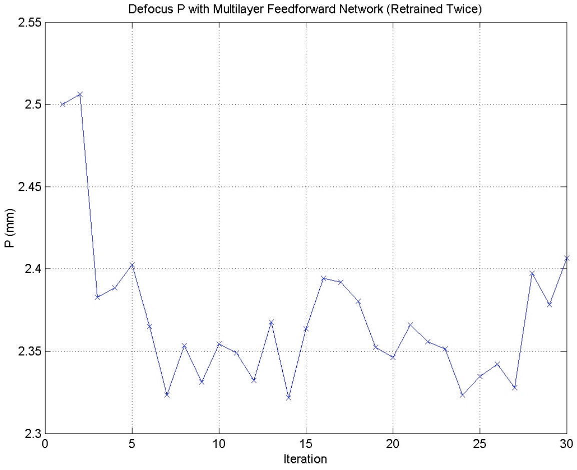

Figure 6.32: Defocus Correction with a Multilayer Feedforward Network after Two Retraining Sessions

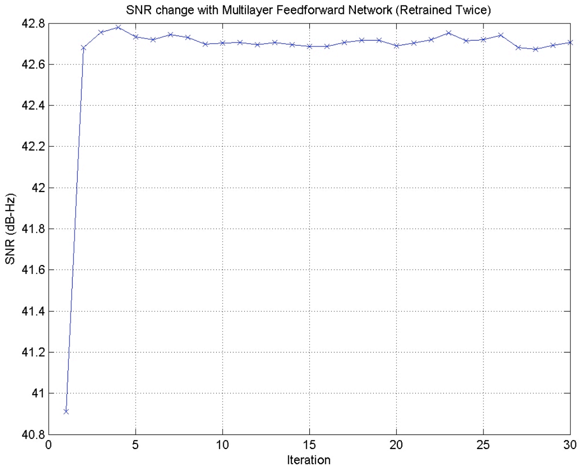

Figure 6.33: SNR as a Function of Time with a Multilayer Feedforward Network after Two Retraining Sessions

In this example, the neural network's adaptability proves to be a key advantage, resulting in a gain in SNR in excess of 2.6 dB-Hz versus the interpolated least squares algorithm. The neural network's adaptability during the course of operation is one of its key advantages over the interpolated least squares algorithm.

6.2.2 Statistical Summary of Acquisition and Tracking Simulations

The true points of maximum SNR at each of the three test elevations are:

1. 15-degrees: (0.50,-0.55,0.20)
2. 45-degrees: (0.20,-0.60,0.70)
3. 75-degrees: (-0.05,1.50,1.15)

The above numbers are accurate to within 0.05-millidegrees in XEL/EL and to within 0.05-millimeters in *P*. Using these offsets, we can calculate mean, standard deviation, and total RMS errors during the tracking process. All tracking tests were performed with the point (0,0,0) set to 40 dB-Hz. In each case, 30 points were taken. In each case, the last 20 points were used to compute statistical errors in steady-state. There is one important note regarding the 75-degree case: due to the severely distorted nature of the focal plane field, it is sometimes possible to find local SNR maxima that are nearly optimal at points away from the global maximum point. This phenomenon will be discussed later in this section.

Table 28 below shows SNR information gathered during the tracking phases, and the true peak SNR is shown in the far right column. It is observed that both multilayer feedforward networks and RBF networks come close to achieving the true SNR peak, often within 0.2 dB-Hz of the peak after retraining at 15 degrees and at 45 degrees θ_{EL}. The cubic spline interpolated least squares algorithm achieves similar performance, but the quadratic interpolated version exhibits greater errors near the optimal point and thus fails to achieve the peak. Although the results of Section 0 indicate that cubic-spline interpolation yields results comparable to quadratic interpolation over a test grid, we see from tracking results that cubic-spline interpolation is the superior method near the origin. At 75 degrees elevation, however, only multilayer feedforward networks with retraining based on real-time data come close to the SNR peak. RBF networks with retraining outperform both interpolated least squares algorithms, but they still fail to achieve the performance of multilayer feedforward networks.

Table 28: Steady-State SNR in dB-Hz

	15-DEG								
	Cubic	**Quad**	**FF 0**	**FF 1**	**FF 2**	**RBF 0**	**RBF 1**	**RBF 2**	**TRUE**
Mean	40.00	38.71	39.98	40.07	40.08	39.99	40.08	40.12	40.16
Std	0.05	0.36	0.09	0.03	0.02	0.06	0.04	0.03	
Low	39.95	38.36	39.88	40.04	40.06	39.94	40.05	40.09	
High	40.04	39.07	40.07	40.10	40.09	40.05	40.12	40.15	
	45-DEG								
	Cubic	**Quad**	**FF 0**	**FF 1**	**FF 2**	**RBF 0**	**RBF 1**	**RBF 2**	
Mean	39.92	35.39	40.22	40.64	40.76	39.87	40.16	40.48	40.85
Std	0.22	0.17	0.27	0.05	0.04	0.22	0.35	0.25	
Low	39.70	35.21	39.95	40.59	40.72	39.64	39.80	40.23	
High	40.14	35.56	40.48	40.69	40.80	40.09	40.51	40.73	
	75-DEG								
	Cubic	**Quad**	**FF 0**	**FF 1**	**FF 2**	**RBF 0**	**RBF 1**	**RBF 2**	
Mean	39.97	39.55	39.96	42.34	42.35	39.99	40.30	41.26	42.87
Std	0.23	0.59	0.13	0.03	0.03	0.10	0.40	0.07	
Low	39.73	38.96	39.83	42.31	42.33	39.90	39.89	41.19	
High	40.20	40.14	40.09	42.37	42.38	40.09	40.70	41.32	

SNR performance is illustrated graphically from Figure 6.34 through Figure 6.36. Here, the neural networks are observed to have a minor performance advantage, due to adaptability, over both versions of interpolated least squares at 15 degrees and at 45 degrees. This advantage becomes significant at 75 degrees elevation as shown in Figure 6.36.

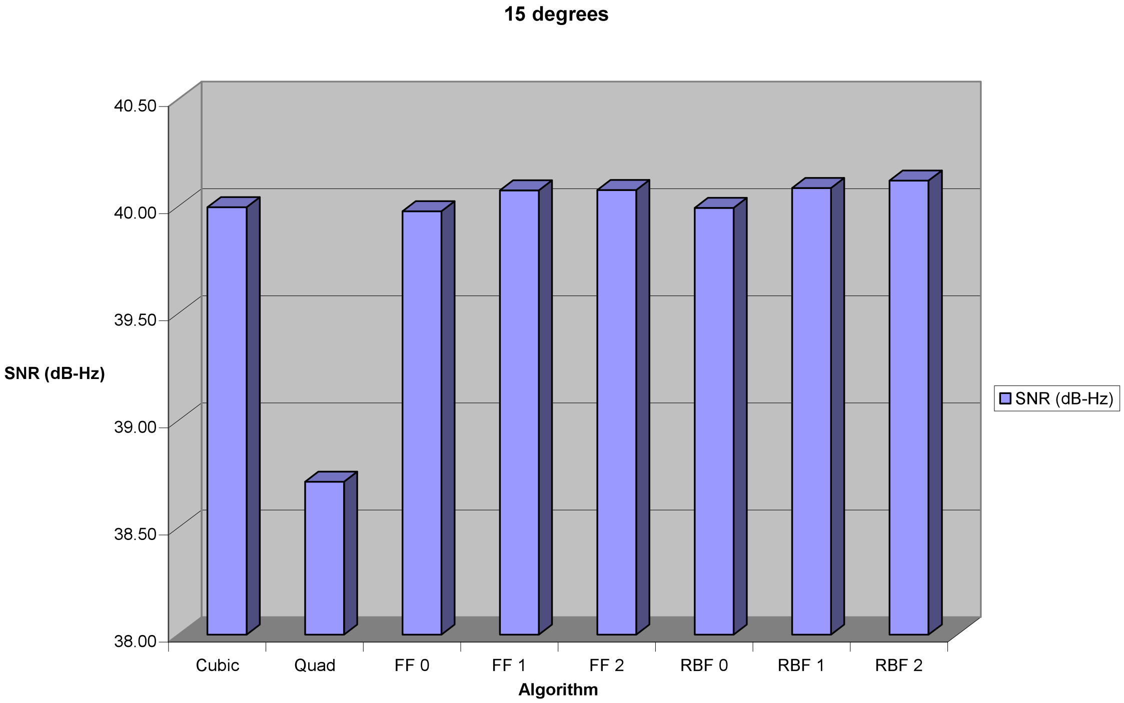

Figure 6.34: SNR Comparison during Tracking at 15 Degrees Elevation

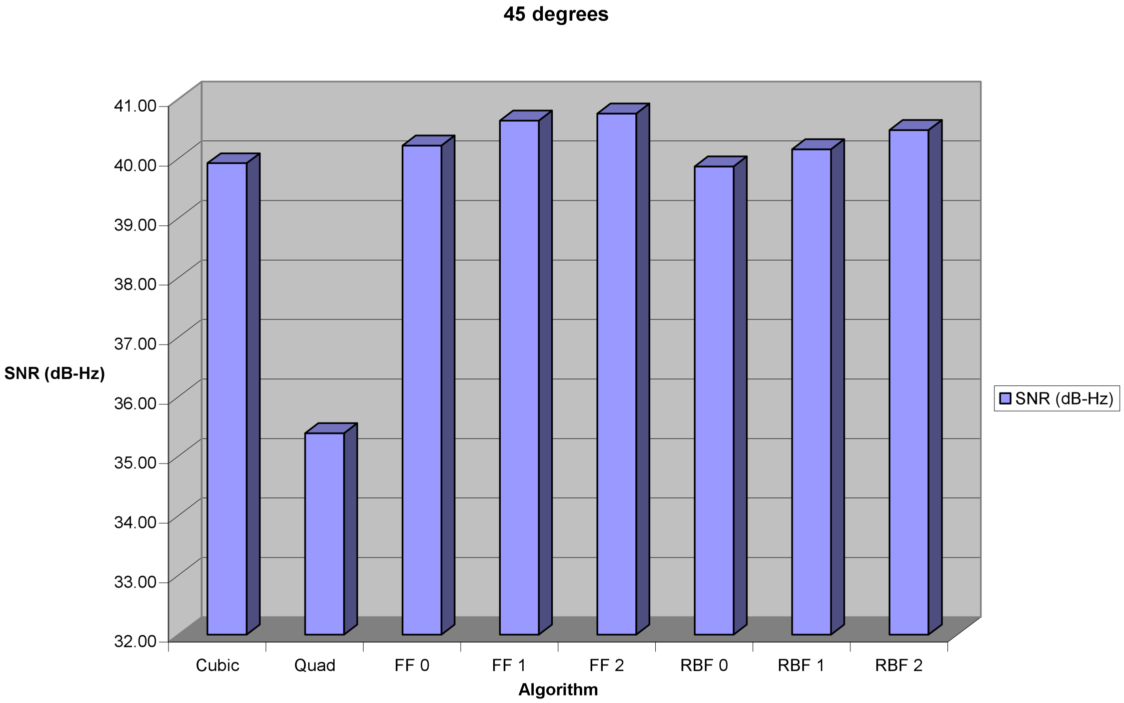

Figure 6.35: SNR Comparison during Tracking at 45 Degrees Elevation

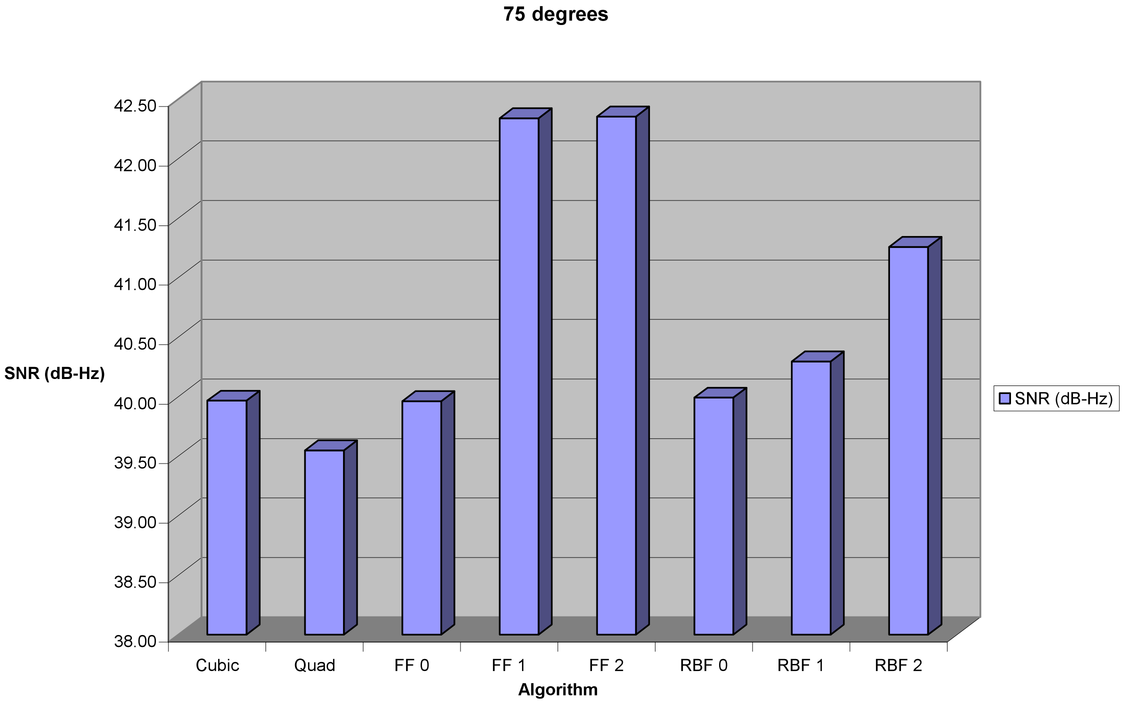

Figure 6.36: SNR Comparison during Tracking at 75 Degrees Elevation

At this stage, it could be argued that the advantage of adaptability in neural networks is a moot advantage. One could argue that the interpolated least squares algorithm could be improved by adding a bias to its output based upon previous experience. This would lead to a change in the nominal center point, the point of highest SNR. Although such a bias scheme can work, it introduces another weakness. Here, we are assigning new ordered triples to each of the points in the lookup table with a bias value added to each of the three coordinates. While this can bias the system toward the true optimal point found during real-time tracking, it also means that we are operating closer to the edges of the lookup table. Operating near the edge can result in reduced interpolation accuracy, especially in the case of cubic spline interpolation. It is desirable instead to have the system learn the region surrounding the true optimal point, and neural networks offer such a capability.

Pointing and defocus correction capability are both shown in Table 29 and Table 30. Multilayer feedforward networks do very well at 15-degrees and at 45-degrees. However, at 75-degrees, there is a large amount of systematic error and relatively little random error. This is especially true of defocus. Here, the mean (systematic) errors exceed 1.0-millimeter while the error standard deviations are low. Although multilayer feedforward networks achieve high SNR performance they still fall short of the true peak. Since the systematic error in defocus is so large, it is reasonable to say that this error is due to the presence of a local maximum that is close, in terms of SNR in dB-Hz, to the true maximum. The result is that the system has taken a strong (almost optimal) local peak to be the global peak and the neural network has been retrained using this spurious peak as its (0,0,0) location.

Table 29: Pointing Errors in Millidegrees during Steady-State Tracking

15-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean XEL	-0.4908	-0.6265	-0.4849	-0.2620	-0.2368	-0.5009	-0.3262	-0.1242
Mean EL	0.5379	0.5684	0.4863	0.1143	-0.0449	0.5128	0.1071	0.1670
Std Dev	0.0599	0.1135	0.1685	0.1839	0.1864	0.1815	0.1909	0.1868
Total RMS	0.7306	0.8535	0.7071	0.3399	0.3047	0.7395	0.3928	0.2797
45-DEG								
Mean XEL	-0.2599	-0.1132	-0.0132	-0.0218	0.0049	-0.1674	-0.1169	0.0320
Mean EL	0.5738	0.5363	0.4735	0.2693	0.3910	0.4307	0.7701	-0.2981
Std Dev	0.2176	0.0883	0.2778	0.3177	0.2969	0.9335	0.8627	1.0091
Total RMS	0.6664	0.5551	0.5491	0.4170	0.4910	1.0416	1.1623	1.0527
75-DEG								
Mean XEL	0.0446	0.0078	0.0376	0.8875	0.7927	0.0558	0.1253	0.1654
Mean EL	-1.4602	-1.5674	-1.5549	-0.6742	-0.5653	-1.5041	-1.3527	-1.2285
Std Dev	0.0882	0.2111	0.0667	0.0609	0.0610	0.0923	0.0956	0.0869
Total RMS	1.4635	1.5816	1.5567	1.1162	0.9755	1.5080	1.3618	1.2426

Table 30: Parabolic Deformation in Millimeters *(P)*

15-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean	-0.2600	-1.4785	-0.6001	0.1451	0.5580	-0.5808	0.2924	0.3790
Std	0.3813	0.2403	0.0673	0.1796	0.0501	0.0559	0.0743	0.0891
Total RMS	0.4615	1.4979	0.6039	0.2309	0.5603	0.5835	0.3017	0.3894
45-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean	-0.8270	-2.6412	-0.2530	0.0451	0.3222	-0.9440	-0.1792	0.0483
Std	0.3409	0.0455	0.1460	0.0858	0.0755	0.1323	0.0834	0.0610
Total RMS	0.8945	2.6416	0.2921	0.0969	0.3310	0.9532	0.1977	0.0778
75-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean	-1.2959	-1.6116	-1.1756	1.3957	1.6037	-1.4638	-0.7526	-0.4416
Std	0.3015	0.0846	0.3534	0.0313	0.0369	0.1044	0.1432	0.0394
Total RMS	1.3305	1.6139	1.2275	1.3960	1.6042	1.4676	0.7661	0.4434

Table 30: Equivalent subreflector position in Millimeters *(Z)*

15-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean	0.6838	3.8885	1.5783	-0.3816	-1.4675	1.5275	-0.7690	-0.9968
Std	1.0028	0.6320	0.1770	0.4723	0.1318	0.1470	0.1954	0.2343
Total RMS	1.2137	3.9395	1.5883	0.6073	1.4736	1.5346	0.7935	1.0241
45-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean	2.1750	6.9464	0.6654	-0.1186	-0.8474	2.4827	0.4713	-0.1270
Std	0.8966	0.1197	0.3840	0.2257	0.1986	0.3479	0.2193	0.1604
Total RMS	2.3525	6.9474	0.7682	0.2548	0.8705	2.5069	0.5200	0.2046
75-DEG	Cubic	Quad	FF 0	FF 1	FF 2	RBF 0	RBF 1	RBF 2
Mean	3.4082	4.2385	3.0918	-3.6707	-4.2177	3.8498	1.9793	1.1614
Std	0.7929	0.2225	0.9294	0.0823	0.0970	0.2746	0.3766	0.1036
Total RMS	3.4992	4.2446	3.2283	3.6715	4.2190	3.8598	2.0148	1.1661

In summary, the adaptability of neural networks permits them to achieve nearly optimal SNR performance even in the presence of a systematic error in the training set. A neural network based system has the ability to use data acquired during real-time operation in order to retrain itself, and this capability enables higher SNR operation to be achieved. The system in present form is not perfect: at 75-degrees it retrained itself to aim for a local peak that was nearly as strong as the global peak, resulting in high systematic errors. Nevertheless, performance is still better than that of the interpolated least squares algorithms in the case of a flawed reference or training set.

6.2.3 The case of excessively large spacing in *P* in the training set

To illustrate the adaptability of multilayer feedforward networks under difficult conditions, we have created a training set with 2.0-millimeter spacing in the parabolic deformation parameter *P* (i.e. 1.0-millimeter spacing in a_2^0). This causes severe performance degradation with the interpolated least squares algorithm. Although a multilayer feedforward network trained on such a flawed set initially fares no better, retraining based on real-time data is seen to yield improved performance. This is another illustration of the usefulness of adaptability. It also illustrates the difficulties encountered by the cubic-spline interpolated least squares algorithm if excessively large table spacings are used.

This simulation example is based on an antenna pointing elevation of 45 degrees. Assume that the antenna starts at the point (+3,-3,-2.5) in *(XEL,EL,P)*-space, which is (+3,-3,+6.58) in *(XEL,EL,Z)*-space, with pointing given in millidegrees and defocus given in millimeters. Performance of the cubic spline interpolated least squares algorithm is shown from Figure 6.37 through Figure 6.39, and the algorithm exhibits oscillatory behavior and poor performance. The performance of the neural network prior to retraining is also poor (Figure 6.40 through Figure 6.42) but there is improvement after the retraining session (Figure 6.43 through Figure 6.45). Although the final 37 dB-Hz SNR in Figure 6.45 is not the 40 dB-Hz we seek, it is also better than the 24- to 26-dB-Hz SNR that the interpolated least squares algorithm (cubic spline version) reached in Figure 6.39. Once the retrained neural network converges, it maintains steady (but suboptimal) tracking, which is better that the oscillatory behavior exhibited by the cubic spline interpolated least squares algorithm.

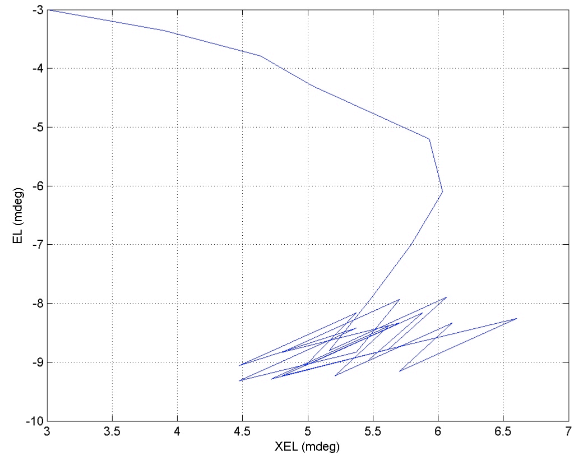

Figure 6.37: Pointing plot at 45 degrees elevation: Interpolated least squares algorithm with cubic spline interpolation

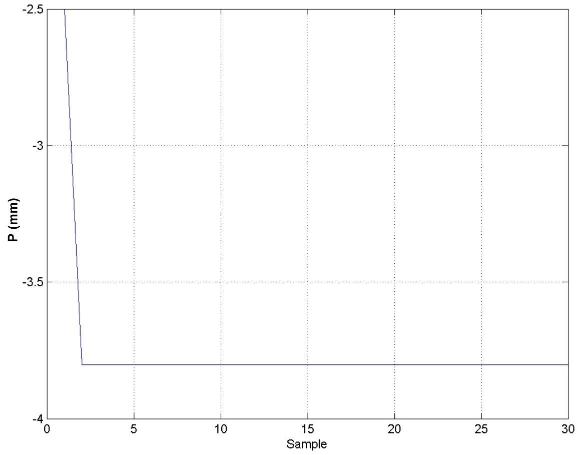

Figure 6.38: Defocus plot at 45 degrees elevation: Interpolated least squares algorithm with cubic spline interpolation

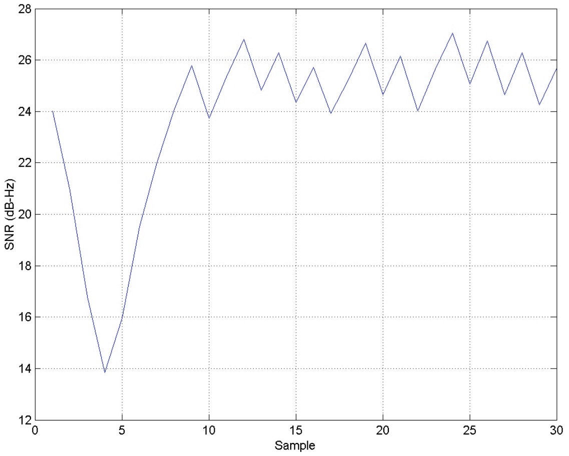

Figure 6.39: SNR plot at 45 degrees elevation: Interpolated least squares algorithm with cubic spline interpolation

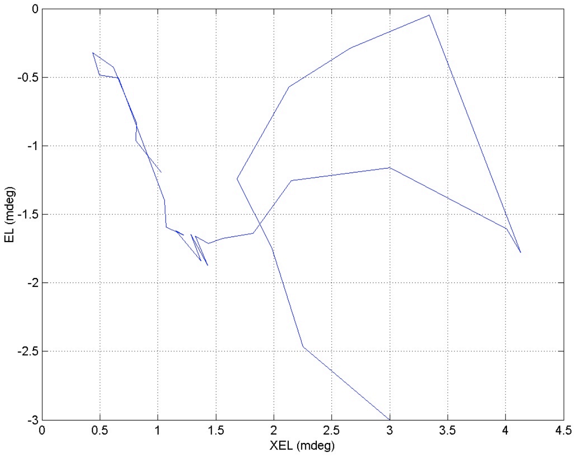

Figure 6.40: Multilayer feedforward network pointing at 45 degrees elevation without retraining

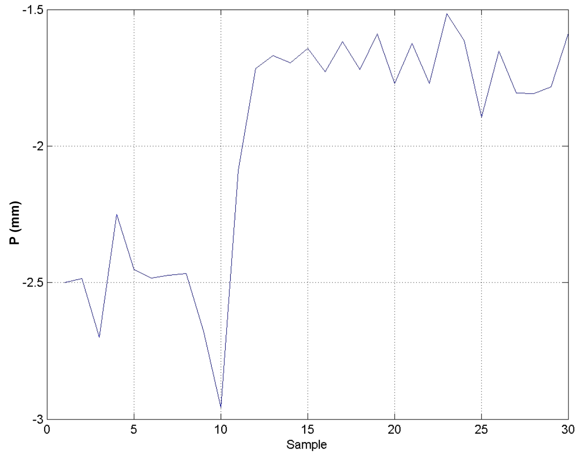

Figure 6.41: Multilayer feedforward network defocus performance at 45 degrees elevation without retraining

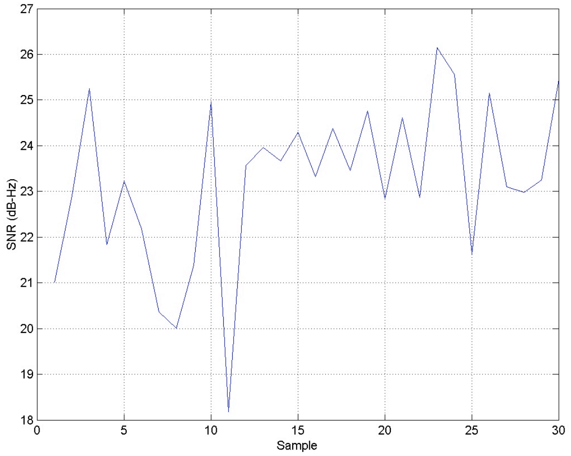

Figure 6.42: Multilayer feedforward network SNR performance at 45 degrees elevation without retraining

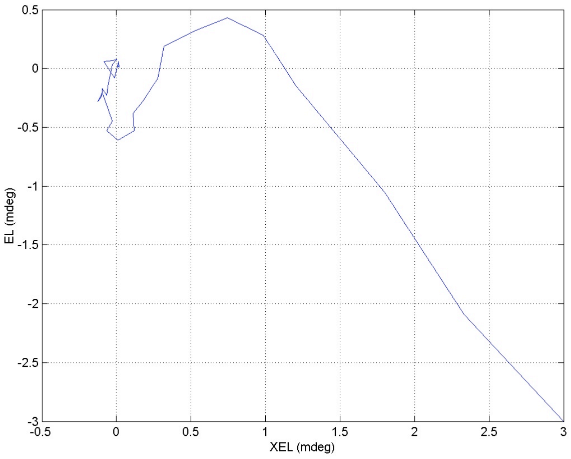

Figure 6.43: Multilayer feedforward network pointing at 45 degrees elevation after first retraining session

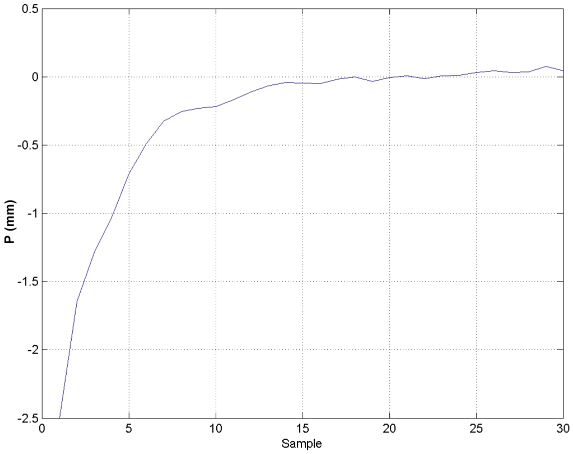

Figure 6.44: Multilayer feedforward network defocus performance at 45 degrees elevation after first retraining session

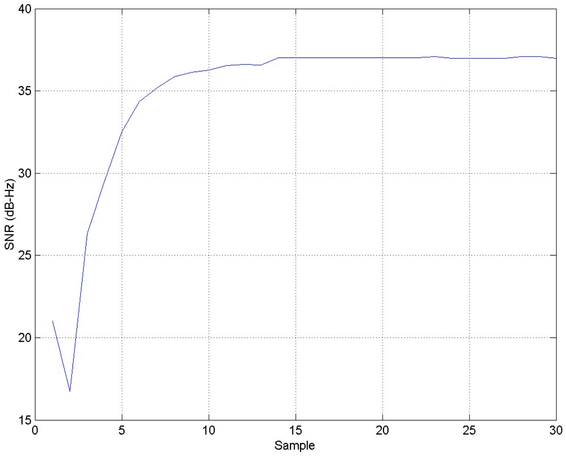

Figure 6.45: Multilayer feedforward network SNR performance at 45 degrees elevation after first retraining session

The preceding is an example of the performance improvement possible with neural network retraining using real-time data. Adding biases to the output of the interpolated least squares algorithm would have yielded a performance improvement in the example illustrated in Section 6.2.1. Here, however, Figure 6.37 shows that there is a notable random variation in interpolated least squares pointing, and this problem is not solved by simply using an additive bias. To correct the interpolated least squares algorithm's behavior, a new, non-aliased raster scan would be necessary. To correct the neural network, by contrast, real-time tracking data can be used without having to gather an entire raster scan.

One possible criticism of this example is that no actual raster scan would involve such large spacing in *P*. This example does not aim at realism: it aims to illustrate neural network adaptability. Here, we observe that neural networks adapt well even if they had originally been trained on a defective training set. Even though excessive spacing is not an issue with a properly designed raster scan pattern, there are other potential imperfections in any raster scan set. By using a severely distorted set as an example, we have illustrated the adaptability advantage of neural networks even under very severe operating conditions.

6.3 Summary of simulation results and discussion

Static tests of interpolated least squares and neural network estimators reveal that overall estimation performance at high SNR is comparable, with neural networks having a minor advantage in static grid tests. At lower SNR, neural network performance falls beneath that of interpolated least squares algorithms in these simulations and in simulations carried out in previous, pointing-only research [28-31].

Acquisition and tracking simulations illustrate two things. With regard to the interpolated least squares algorithm, cubic spline interpolation yields superb accuracy near the origin of pointing-defocus space, but quadratic interpolation does not show large changes in accuracy as we move away from the center as cubic spline interpolation does. This means that while the two have similar accuracy over static test grids, cubic-spline interpolation yields better tracking accuracy in practice and is a good benchmark against which neural networks can be compared. With regard to neural networks, accuracy near the center of training space is also high, and this is likely due to the fact that the center is surrounded by many neighboring points used in training. With a greater variety of neighboring training examples near the center, a neural network can often achieve better accuracy near the center than at the edges of the training region. Indeed, tracking tests show accuracy that is almost identical to that achieved by cubic-spline interpolated least squares algorithms. As stated in Section 4.6, acquisition and tracking are relatively robust with respect to estimation errors under most conditions, and estimation accuracy near the origin is most important. This is reflected in the acquisition and tracking simulation results presented in this chapter. We now turn our attention to real-world experimental results.

7 Real-World Experimental Results

A number of real-world experiments involving the quadratic-interpolated least squares algorithm and neural networks were performed at Goldstone, CA. While the experiments at the 70-meter antenna at DSS-14 involved only pointing correction and are also documented in [31] along with some pointing only simulations, the experiments at the 34-meter antenna at DSS-13 involved both pointing and defocus correction. As stated in [31], RBF networks achieve accuracy close to that of the interpolated least squares algorithms in the pointing-only case. Multilayer feedforward networks were used in the subsequent pointing and defocus tests at the 34-meter antenna at DSS-13. At both antennas, the quadratic interpolated least squares algorithm was used as a "benchmark" for comparison.

Before we proceed, we note that in Section 7.2 the defocus parameter of interest is not *P* (parabolic deformation) as it was in Chapter 6 but is rather *Z*, the subreflector's position along the *z*-axis. This is a parameter over which we had direct control and, in real-world situations, it is the parameter of interest. Although most of our results are quoted in terms of *Z*, we present them in equivalent terms of parabolic deformation *P* in the summary table at the end of Section 7.2. In the case of the 34-meter antenna, we note that a +1.0-millimeter change in *P* is equivalent to a -1.89-millimeter change in *Z*.

7.1 Experimental Data from the 70-meter antenna at DSS-14

7.1.1 RBF Network Pointing Results

Reference data sets were collected for the pointing-only case on days 52 and 62 of the year 2001. Each data set consisted of 49 points in the *(XEL,EL)*-plane spaced 1.0-millidegree apart and ranging from -3.0 to +3.0-millidegrees in both *XEL*- and *EL*-directions. These data were used both as reference tables for the interpolated least squares algorithm and as training sets for RBF networks.

Acquisition and tracking were performed using RBF networks on day 68 of 2001 at DSS-14. In each case, a pointing offset was intentionally applied to the antenna, and RBF networks were used to correct the pointing offset and to maintain steady-state pointing.

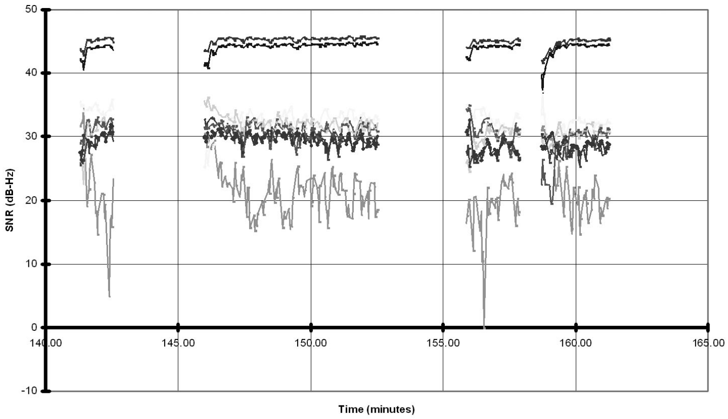

Figure 7.1: Day 68 acquisition and tracking example.

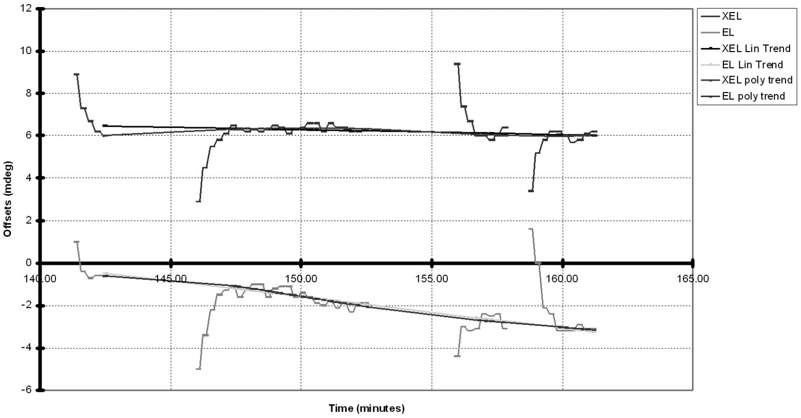

Figure 7.2: Day 68 acquisition and tracking example. © 2002 IEEE. Reprinted with permission from Mukai, R., Vilnrotter, V.A., Arabshahi, P., and Jamnejad, V., "Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas", ***IEEE Transactions on Neural Networks*****, vol. 13, no. 5, pp. 1149-1162, Sep, 2002**

As seen from Figure 7.1 and Figure 7.2, RBF networks are able to both correct pointing errors and maintain steady-pointing, keeping RMS errors within DSN requirements. In these two figures, the antenna was descending in pointing elevation from 62.7 degrees to 59.2 degrees. As shown in Figure 7.2, four different pointing offsets were applied to the antenna. In each case, the RBF networks were able to bring the antenna back on point quickly, recovering SNR losses due to the intentionally applied pointing errors as shown in Figure 7.1. In Figure 7.2, the long-term trend lines describing spacecraft position in *(XEL,EL)*-space relative to the antenna's pointing predict model are shown. Here the point (0,0) is taken relative to the antenna's pointing predict model and *not* relative to the spacecraft's position. RMS pointing errors for the steady-state portions of the track without RMS trend-line subtraction were 0.8104-millidegrees. If the linear trend-line is properly subtracted, then steady-state RMS pointing errors are just 0.3179-millidegrees, which is within DSN pointing accuracy requirements. The strongest evidence for the validity of the trend-line comes from the fact that points along the trend-line yield the maximum SNR achieved during the course of operations even though a wide range of pointing offsets surrounding the source had been tested.

One of the four acquisition and tracking examples from this set is shown in Figure 7.3 and in Figure 7.4. These two figures focus on the second acquisition and tracking run illustrated in Figure 7.1 and Figure 7.2. In Figure 7.4 we have taken the point (0,0) relative to the spacecraft and *not* relative to the antenna's pointing predict model as was done in Figure 7.2. For the steady-state portion of this track, we found RMS pointing errors of 0.4219-millidegrees without trend-line subtraction. If the linear trend-line is subtracted, steady-state pointing errors are 0.3074-millidegrees. Figure 7.3 shows that the central channel recovers approximately 3-dB as a result of pointing correction. The combined channel, which is more robust with respect to pointing errors due to the presence of the outer horns, shows an SNR recovery of approximately 2-dB. SNR is seen to remain steady, within measurement error, once the RBF networks have acquired the source.

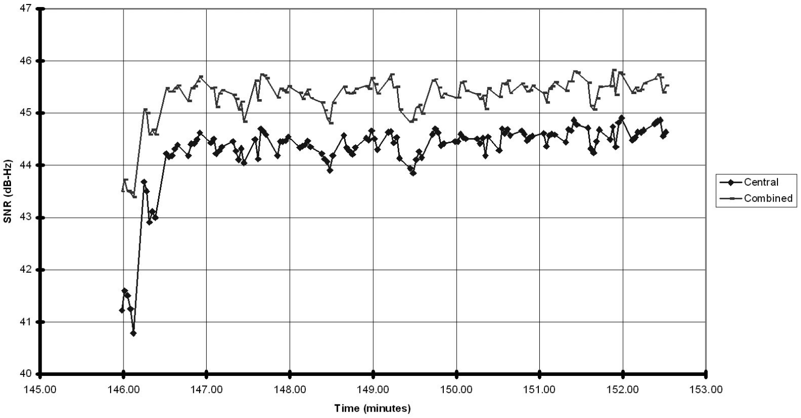

Figure 7.3: Day 68 acquisition and tracking example.

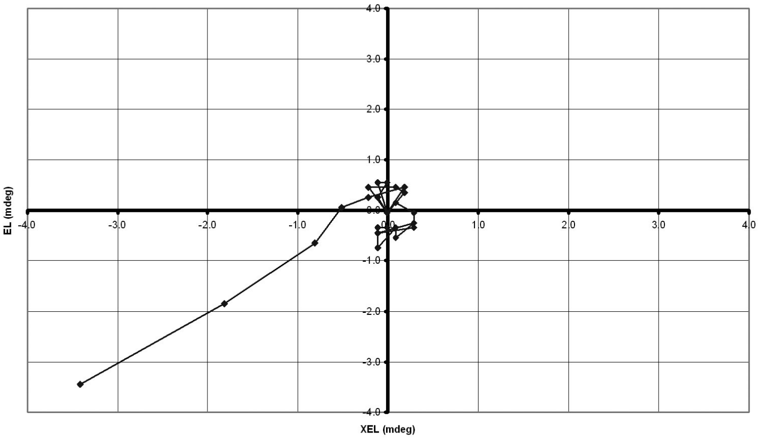

Figure 7.4: Day 68 acquisition and tracking example.

On day 124, RBF acquisition and tracking tests were performed again. The track illustrated from Figure 7.5 through Figure 7.7 shows a steady-state tracking accuracy of 0.3775-millidegrees with the linear trend-line removed and 0.4441-millidegrees without linear trend removal. All plots of offsets for this track were generated relative to the spacecraft. Here, the antenna was descending in elevation from 40.3 degrees to 39.0 degrees. Figure 7.7 illustrates the steadiness of the received SNR once acquisition was complete, with only a small dip occurring during that period. The same figure illustrates a gain of approximately 5-dB due to pointing correction during the initial acquisition phase from a large pointing offset of over 5-millidegrees. While the RBF networks were originally trained over a range of ±3.0-millidegrees they were able to perform successful acquisition well outside of the original training range, indicating their ability to generalize well to novel conditions. A second RBF example from day 124 is illustrated by Figure 7.8 and Figure 7.9. In those two figures, the antenna is descending from 36.0 to 34.6 degrees in pointing elevation. Without linear trend-line removal RMS pointing errors were measured at 0.5154-millidegrees. With linear trend-line removal, RMS pointing errors were measured at 0.4784-millidegrees.

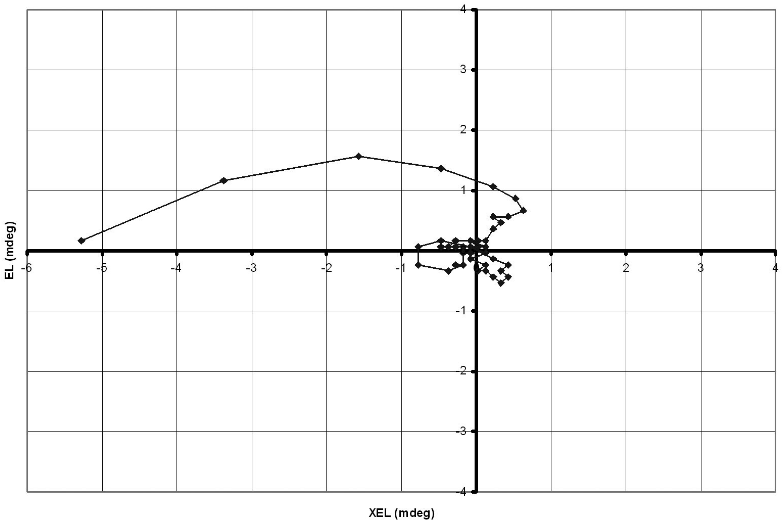

Figure 7.5: Day 124 RBF acquisition and tracking example.

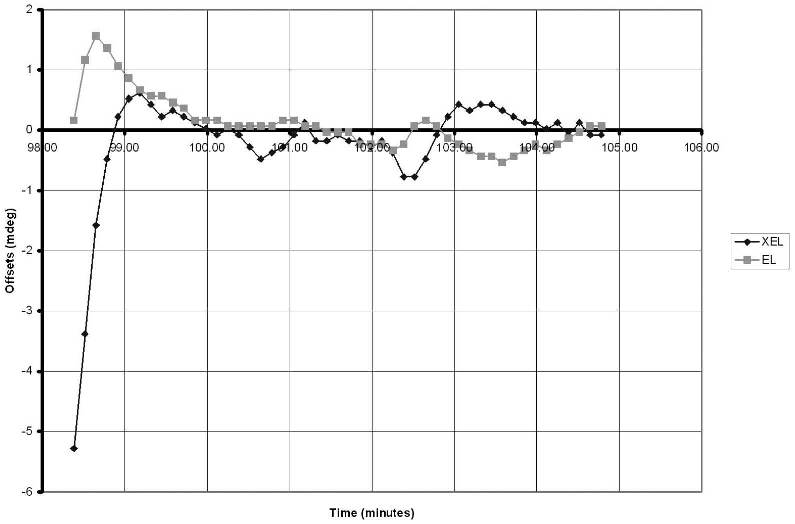

Figure 7.6: Day 124 acquisition and tracking from 40.3 down to 39.0 degrees elevation.

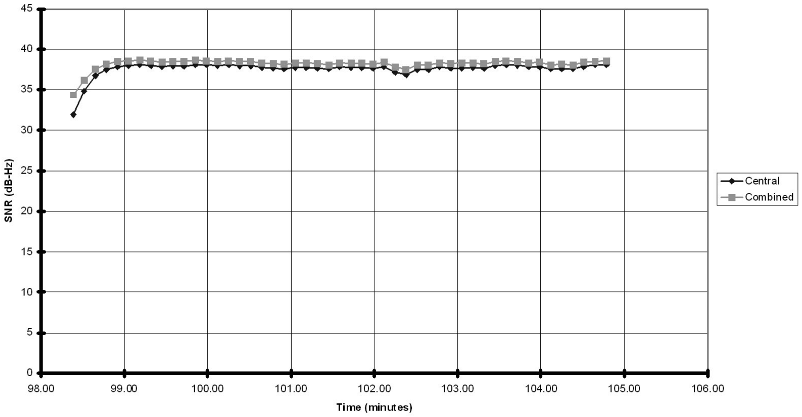

Figure 7.7: Day 124 illustration of SNR recovery as a result of pointing correction.

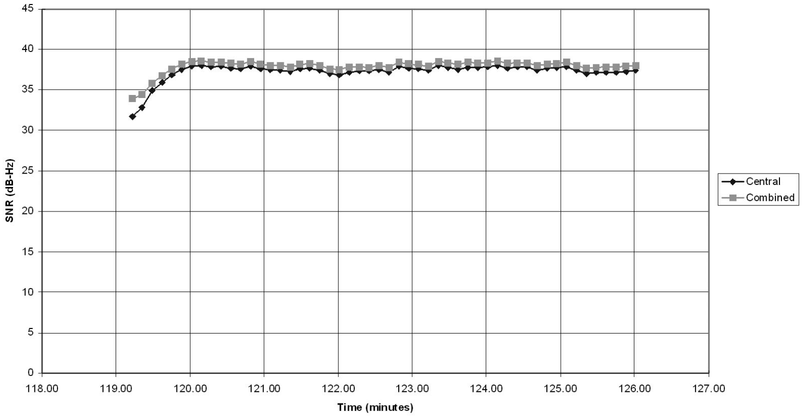

Figure 7.8: Day 124 RBF acquisition and tracking test. The antenna was descending from 36.0 to 34.6 degrees in elevation. We observe an SNR recovery of approximately 5-dB as a result of pointing correction.

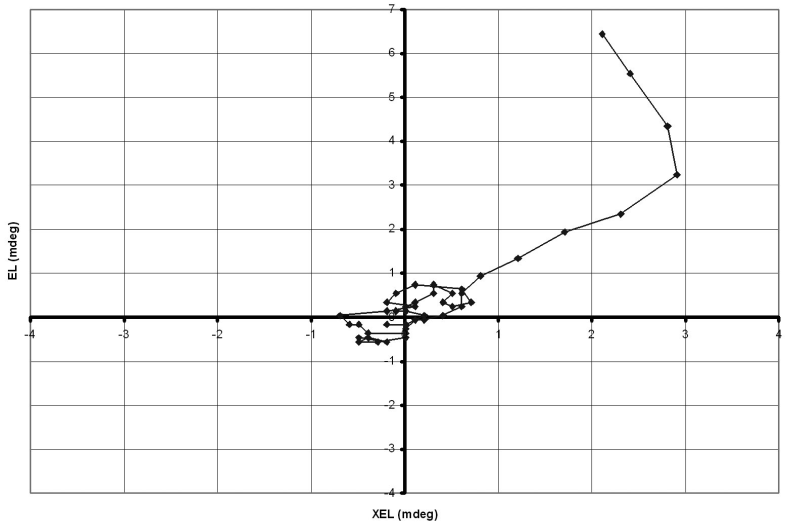

Figure 7.9: Day 124 track with RBF networks. This is the same track as that illustrated by Figure 7.8.

Figure 7.10 and Figure 7.11 illustrate an acquisition and tracking test performed on day 171. The antenna was descending from 41.0 to 39.4 degrees in elevation. Pointing correction yielded SNR recovery of approximately 2.5 to 3.0 dB.

In addition, steady-state tracking accuracy was 0.6428-millidegrees RMS without linear trend-line removal and 0.6334-millidegrees RMS with linear trend-line removal. As with day 124, a pointing offset larger than that used in training was applied in order to test the RBF network's generalization capabilities.

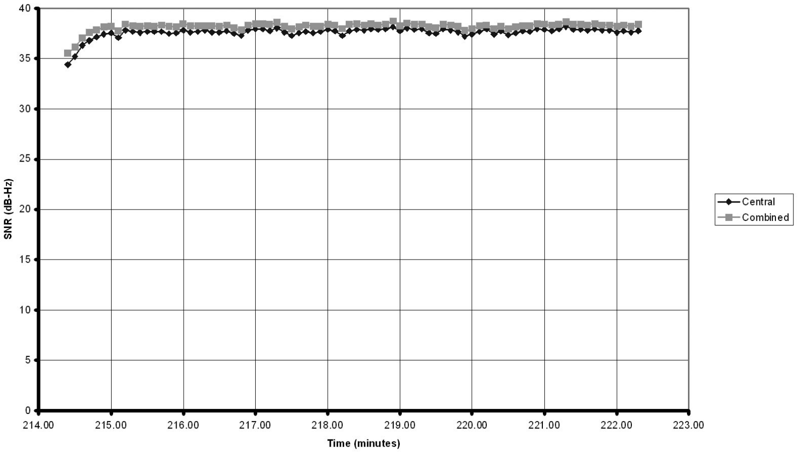

Figure 7.10: Day 171 acquisition and tracking SNR plot.

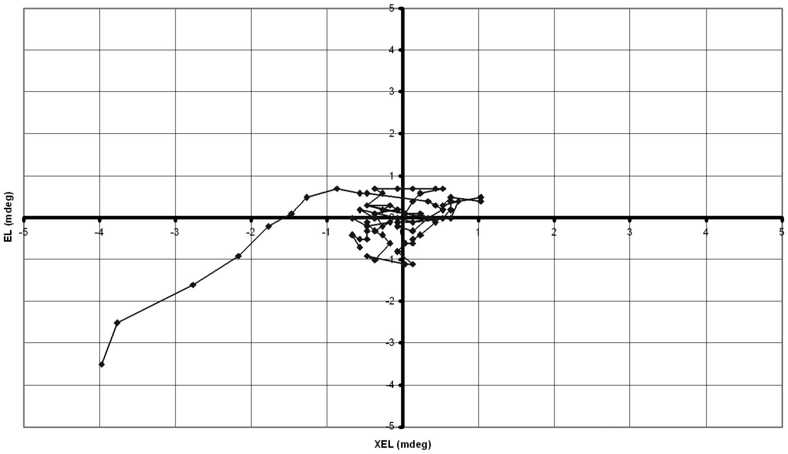

Figure 7.11: Acquisition and tracking plot corresponding to the SNR plot of Figure 7.10 for day 171.

Another day 171 acquisition and tracking test, with the antenna descending from 39.3 to 37.8 degrees in pointing elevation, is shown in Figure 7.12 and Figure 7.13. RMS pointing errors were 0.6521-millidegrees RMS without trend-line removal and 0.5798-millidegrees with trend-line removal.

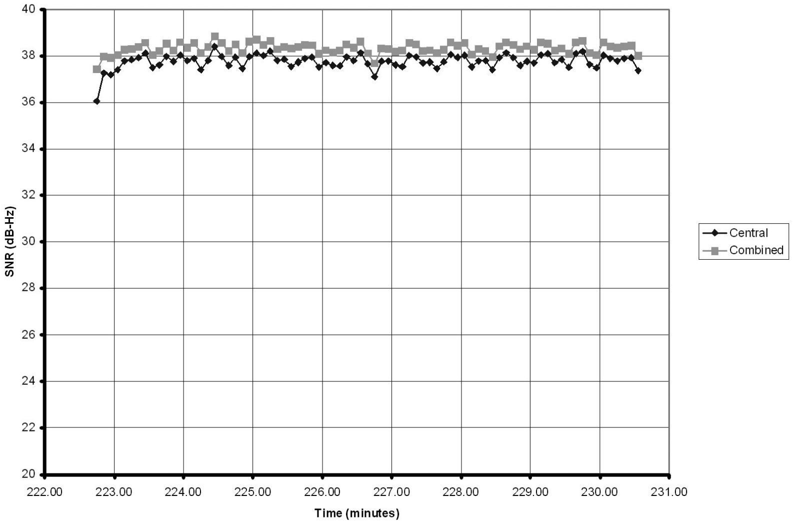

Figure 7.12: Acquisition and tracking SNR plot for day 171. The antenna was descending from 39.3 to 37.8 degrees. This track took place immediately after the track illustrated in Figure 7.10 and Figure 7.11.

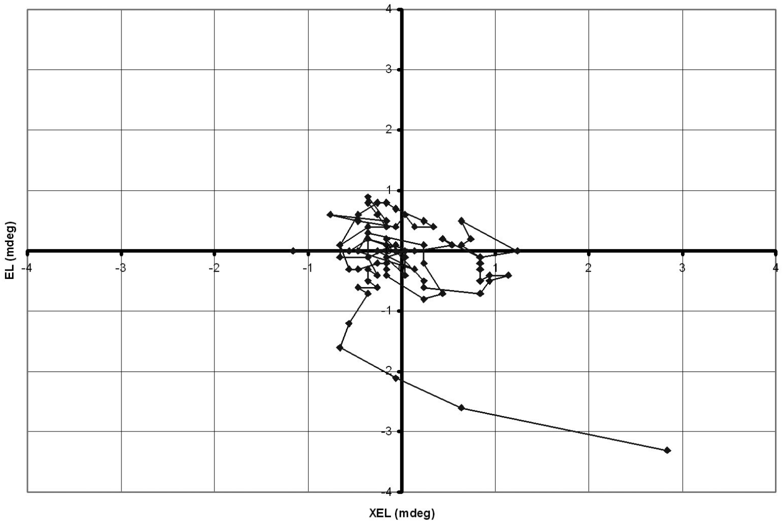

Figure 7.13: The acquisition and tracking plot corresponding to the track of Figure 7.12 (day 171 with the antenna descending from 39.3 degrees to 37.8 degrees elevation).

The first day 195 example illustrates performance with the antenna rising from 35.3 to 37.0 degrees in pointing elevation as illustrated in Figure 7.14 through Figure 7.16. Without trendline removal the RMS pointing error is 0.6242-millidegrees. With trend-line removal, this becomes 0.5006-millidegrees. There is an SNR gain of approximately 1.5 dB-Hz in Figure 7.16 due to pointing correction.

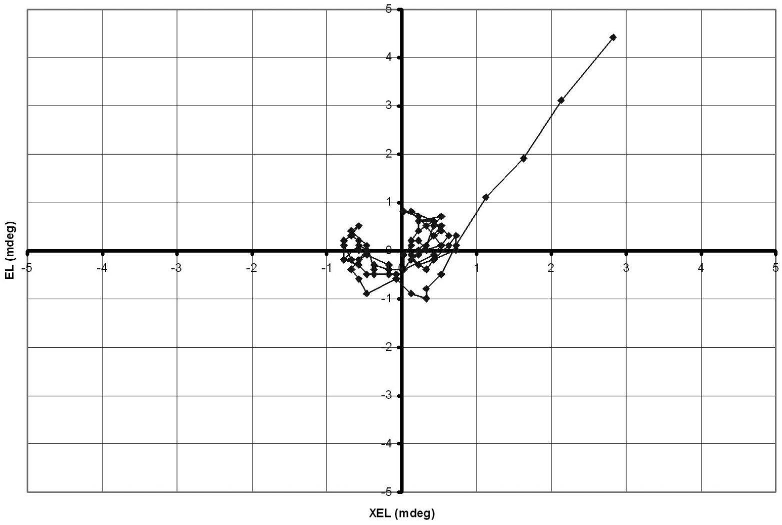

Figure 7.14: Day 195 RBF track with the antenna rising in elevation from 35.3 to 37.0 degrees.

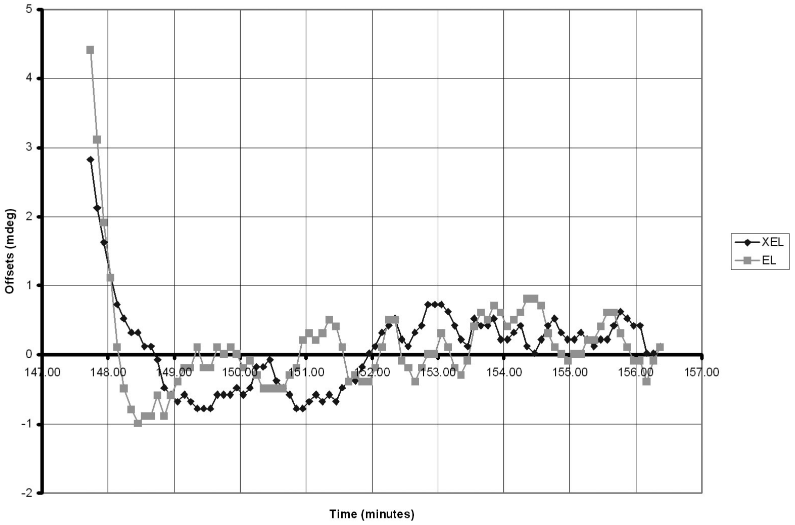

Figure 7.15: Day 195 RBF track with the antenna rising in elevation from 35.3 to 37.0 degrees. This is a time-series plot of the data in Figure 7.14.

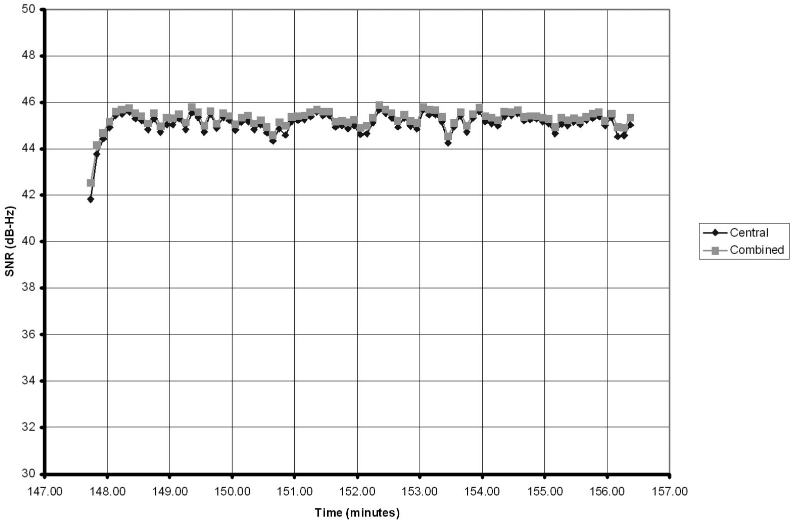

Figure 7.16: Day 195 RBF track with the antenna rising in elevation from 35.3 to 37.0 degrees. This is the same track as illustrated in both Figure 7.14 and Figure 7.15.

Another Day 195 track, illustrated from Figure 7.17 through Figure 7.19 was performed with the antenna rising in elevation from 39.5 degrees to 41.6 degrees. No acquisition from a pointing offset was performed, but steady-state tracking performance was observed over a period of roughly ten minutes. Here, pointing accuracy is very impressive. Even without trend-line removal, errors were only 0.3349-millidegrees RMS, and trend-line removal yielded RMS errors of 0.3179-millidegrees. Figure 7.18 illustrates steady SNR performance during the course of the track.

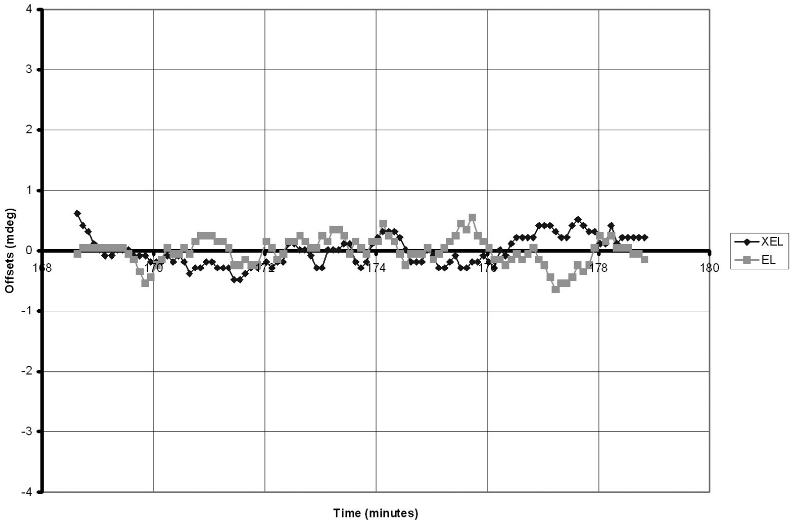

Figure 7.17: Day 195 RBF network track with the antenna rising from 39.5 to 41.6 degrees pointing elevation.

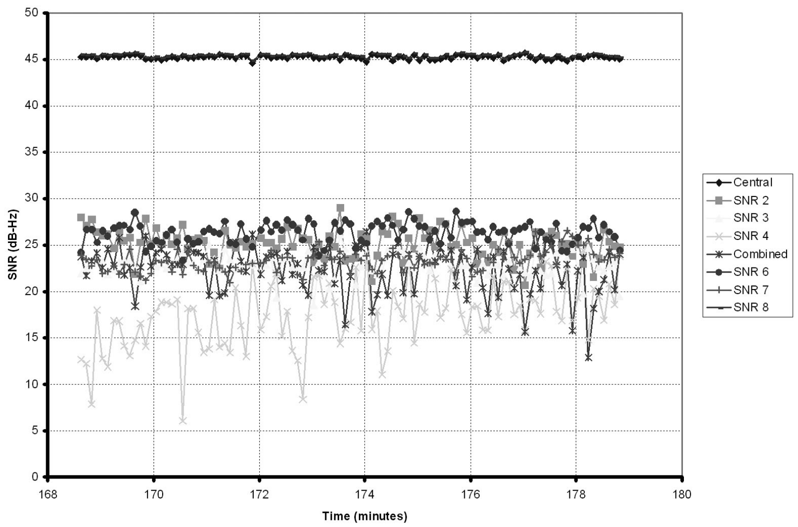

Figure 7.18: Day 195 RBF network track. Antenna was rising in pointing elevation from 39.5 to 41.6 degrees. © 2002 IEEE. Reprinted with permission from Mukai, R., Vilnrotter, V.A., Arabshahi, P., and Jamnejad, V., "Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas", ***IEEE Transactions on Neural Networks*****, vol. 13, no. 5, pp. 1149-1162, Sep, 2002**

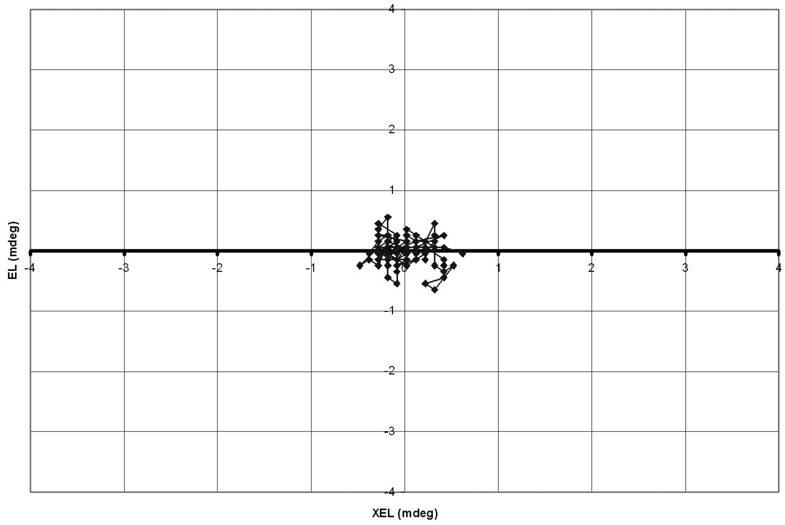

Figure 7.19: Day 195 RBF network track. Antenna was rising in pointing elevation from 39.5 to 41.6 degrees. RMS pointing error, with no long-term trend-line removed, is 0.3349-millidegrees total. If a trend-line is removed, the RMS error estimate is 0.3179 millidegrees total, which is essentially the same.

Another RBF network example, illustrated in Figure 7.20 and Figure 7.21, is an acquisition and tracking example with the antenna rising in elevation from 43.4 to 46.3 degrees. An SNR recovery of approximately 2 dB can be seen in Figure 7.20, with steady tracking SNR maintained after the initial acquisition stage. Steady-state tracking accuracy, illustrated in Figure 7.21, is 0.4984-millidegrees RMS without trend-line removal. With trend-line removal, it is 0.4157-millidegrees RMS.

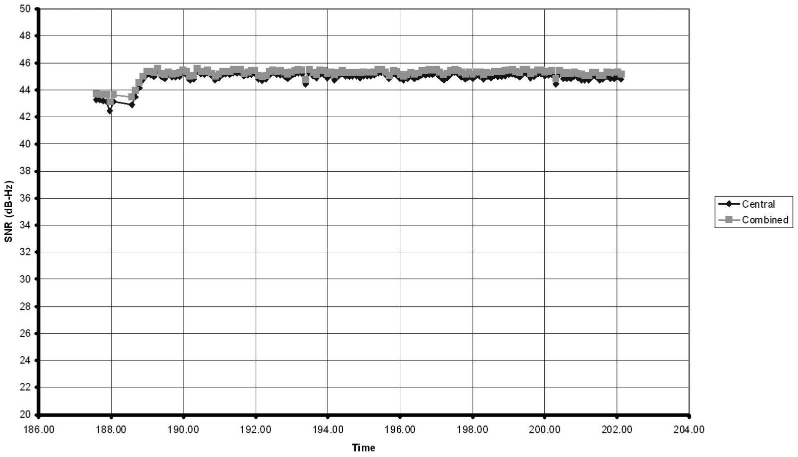

Figure 7.20: Day 195 RBF network track. Antenna was rising in elevation from 43.4 to 46.3 degrees.

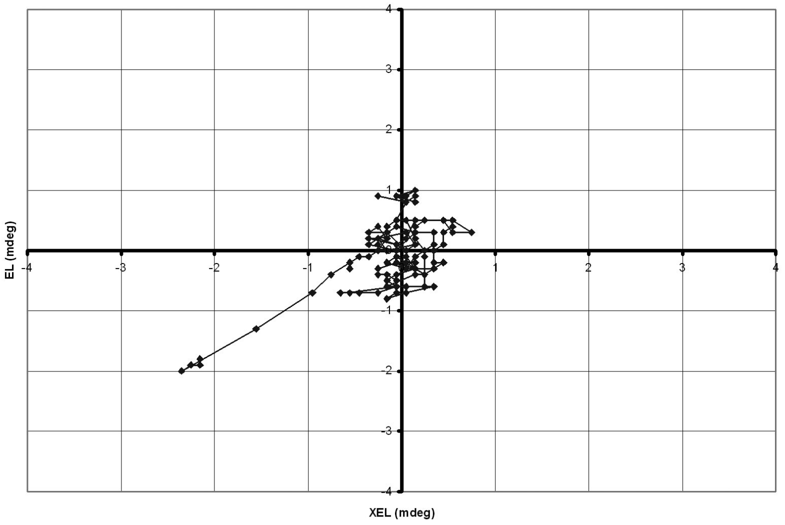

Figure 7.21: Day 195 RBF network track with the antenna rising from 43.4 to 46.3 degrees.

The last RBF network example is a steady-state tracking example without acquisition. The antenna was rising in elevation from 47.9 to 49.9 degrees. RMS pointing errors were 0.7007-millidegrees without trend-line removal and 0.6281-millidegrees with trend-line removal. SNR performance is illustrated in Figure 7.22, and pointing performance is illustrated in Figure 7.23.

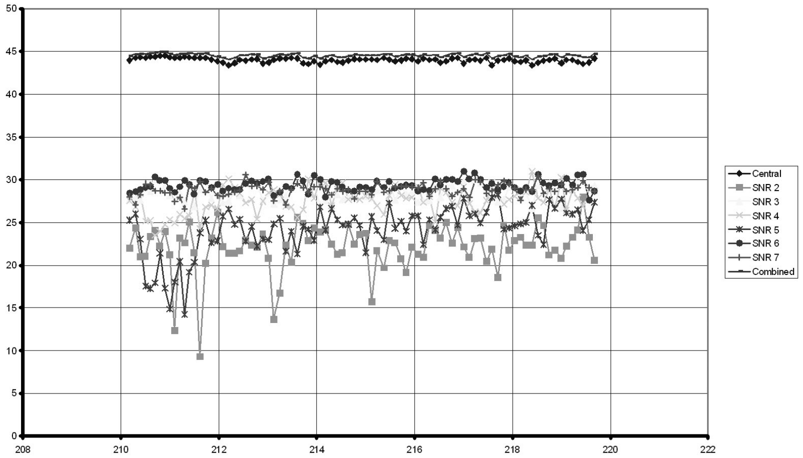

Figure 7.22: Day 195 RBF network track with the antenna rising from 47.9 to 49.9 degrees in elevation.

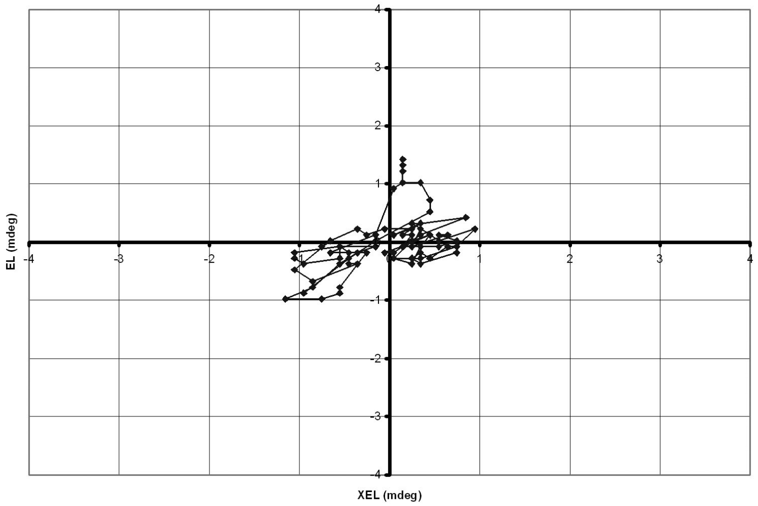

Figure 7.23: Day 195 RBF network with the antenna rising from 47.9 to 49.9 degrees.

7.1.2 Interpolated Least Squares Pointing Results

The quadratic interpolated least squares algorithm was also tested for the pointing only case at DSS-14. Data gathered on days 52 and 62 were used to create the two-dimensional lookup tables used by this algorithm, and the first acquisition and tracking experiments were carried out on day 107.

A series of acquisition tests, combined with brief tracking, were performed on day 107 as shown from Figure 7.24 through Figure 7.27. In this set of five acquisition and tracking tests, the antenna was descending from 41.1 to 39.2 degrees in pointing elevation. An overview of SNR recovery is illustrated in Figure 7.24 and in Figure 7.25, where we see that much of the SNR lost due to intentionally applied pointing errors is quickly recovered. We note changes in the distribution of power in the six outer horns, particularly in the fifth acquisition and track test illustrated in Figure 7.25. In that case, a significant amount of power was captured in one of the outer horns. Correction of antenna pointing resulted not only in a gain in SNR for both the central and combined channels but also a drop in the SNR of that outer horn, indicating a shift of the power to the central horn as desired.

From Figure 7.24, the central channel at the start of the fifth acquisition and tracking test in the figure has lost about 10 dB of SNR due to the pointing error. The loss in the combined channel is less severe and is consistent with the fact that the combined channel would include input from the six outer horns, including an outer horn with a relatively high SNR as shown in Figure 7.25. The centering of the power that is illustrated in Figure 7.25 is good evidence of the accurate re-pointing of the antenna from a large pointing offset in excess of five millidegrees shown in Figure 7.26. Figure 7.26 shows pointing offsets relative to the antenna’s pointing predict model. By contrast, Figure 7.27 shows pointing offsets relative to the mean position of the spacecraft during the tracking phases and thus has its coordinate axes centered on the spacecraft's mean position. The tracking-only phases yield an RMS pointing error of 0.6671-millidegrees without removal of the linear trend line. With trend line removal, RMS pointing accuracy is 0.6505-millidegrees.

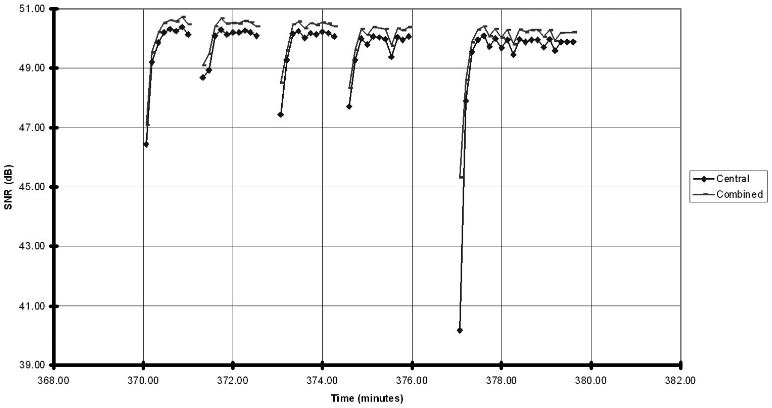

Figure 7.24: Day 107 acquisition and tracking series SNR plot for central and combined channels.

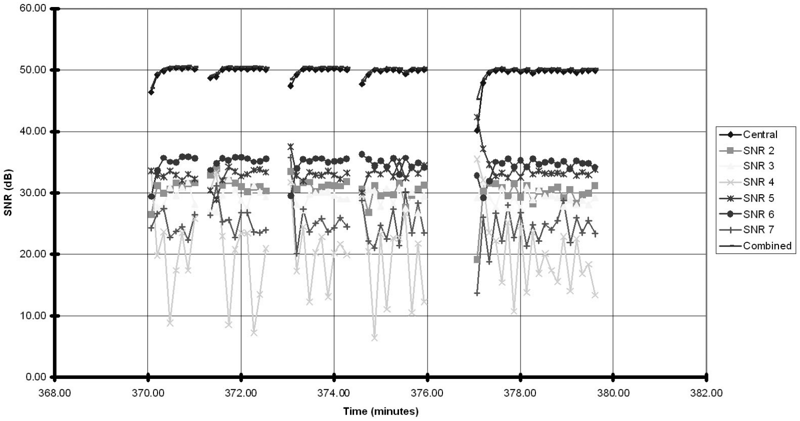

Figure 7.25: Day 107 acquisition and tracking series with the antenna descending from 41.1 to 39.2 degrees in elevation showing SNR of all seven horns plus the combined channel.

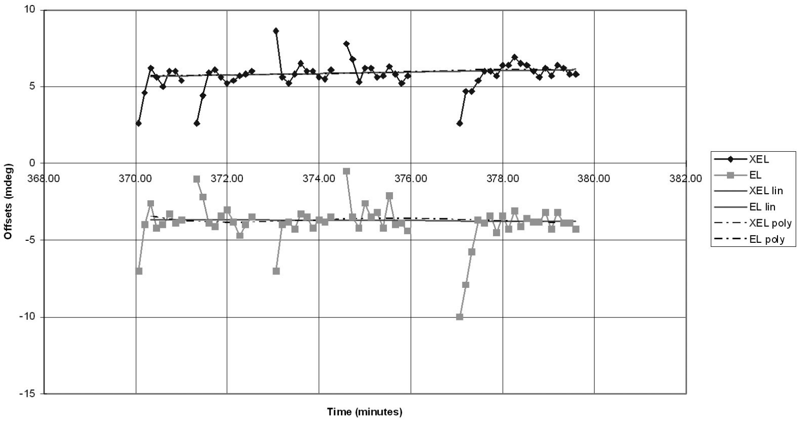

Figure 7.26: Day 107 acquisition and tracking series. Pointing offsets are shown relative to the antenna's pointing predict model. © 2002 IEEE. Reprinted with permission from Mukai, R., Vilnrotter, V.A., Arabshahi, P., and Jamnejad, V., "Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas", *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1149-1162, Sep, 2002

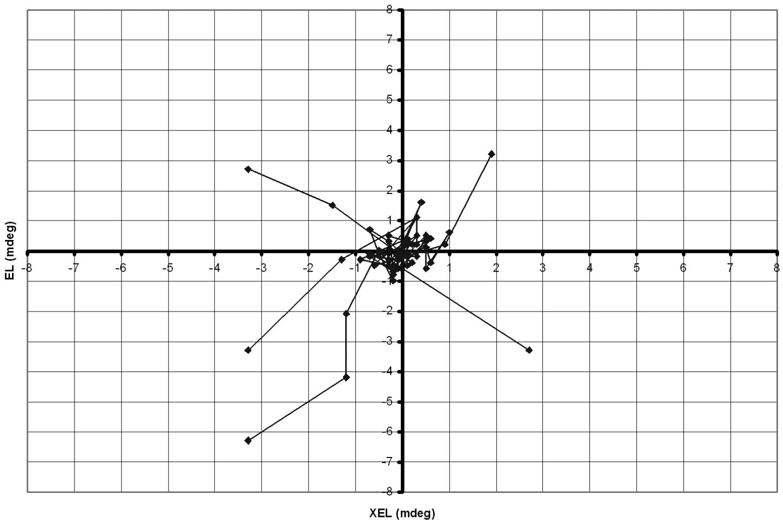

Figure 7.27: Day 107 acquisition and tracking sequence with the antenna descending from 41.1 to 39.2 degrees elevation. These are the pointing offsets illustrated with respect to the spacecraft itself.

An interpolated least squares track from day 167 of 2001 further illustrates the performance of this algorithm. This acquisition and tracking test was carried out with the antenna rising from 34.8 to 37.3 degrees in elevation and is illustrated from Figure 7.28 through Figure 7.31. SNR recovery of 2.5-3.0 dB in the central and combined channels is shown in Figure 7.28. By looking at side-horn SNR data in Figure 7.29, we see that horn 4 has a relatively high SNR while horn 6 has a relatively low SNR at the start of the acquisition period. The SNR becomes somewhat more balanced once acquisition completes, indicating that the power in the focal plane field is better centered on the central horn after acquisition. Pointing performance is illustrated in Figure 7.30 and Figure 7.31. RMS pointing errors are 0.4230-millidegrees without trend-line removal. With trend-line removal, we obtain RMS pointing errors of 0.4141-millidegrees.

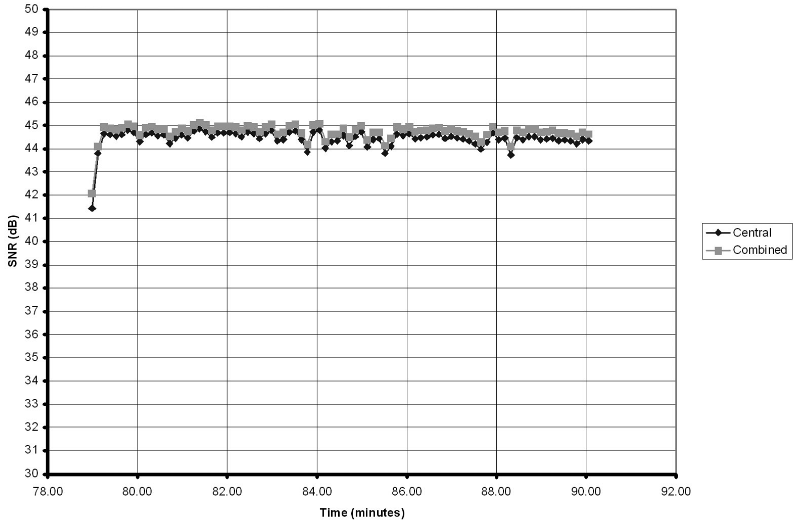

Figure 7.28: Day 167 interpolated least squares track with the antenna rising from 34.8 to 37.3 degrees elevation. SNR recovery of approximately 2.5-3.0 dB is in evidence.

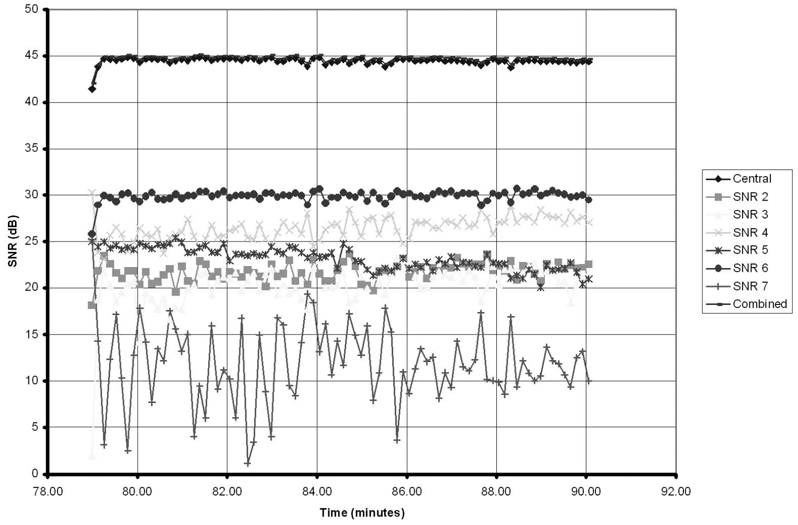

Figure 7.29: Day 167 interpolated least squares track with the antenna rising from 34.8 to 37.3 degrees elevation. Here, the SNR of each of the channels (and of the combined channel) is plotted as a function of time.

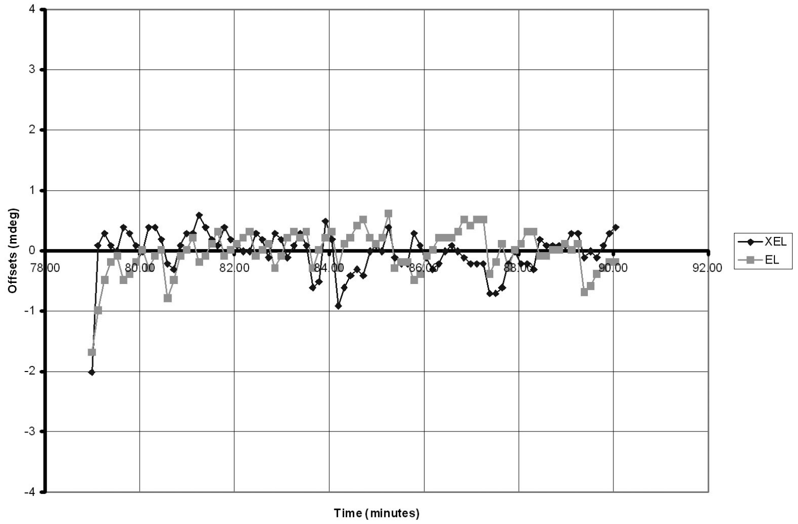

Figure 7.30: Pointing offsets during day 167 interpolated least squares track with the antenna rising from 34.8 to 37.3 degrees elevation.

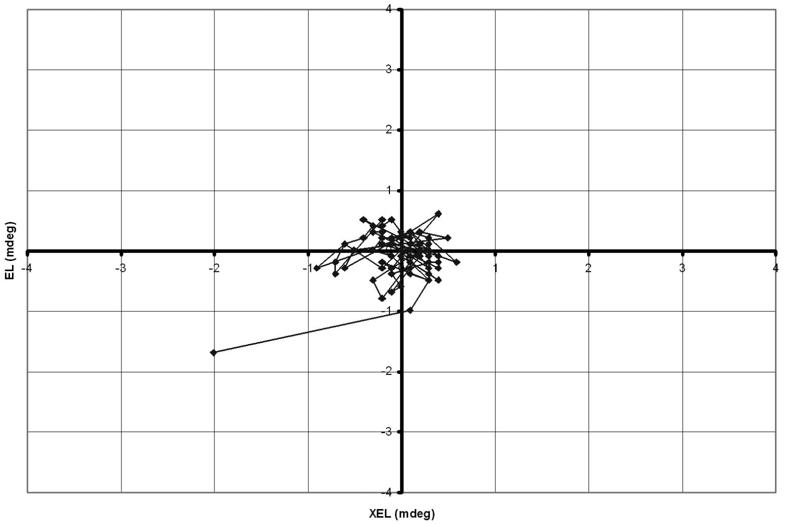

Figure 7.31: Day 167 track with the antenna rising from 34.8 to 37.3 degrees elevation. This illustrates acquisition and tracking in the *(XEL,EL)*-plane.

We also include an interpolated least squares track from Day 195, the last day of experiments at DSS-14. Although no acquisition was performed here, steady-state tracking performance was good, as illustrated from Figure 7.32 through Figure 7.34. Elevation was rising from 49.9 degrees to 51.4 degrees, and this was performed on a warm summer day although the reference table data were gathered on a cold winter night. RMS pointing errors were 0.7648-milldegrees with no trend line subtraction and 0.6380-millidegrees with the trend line subtracted.

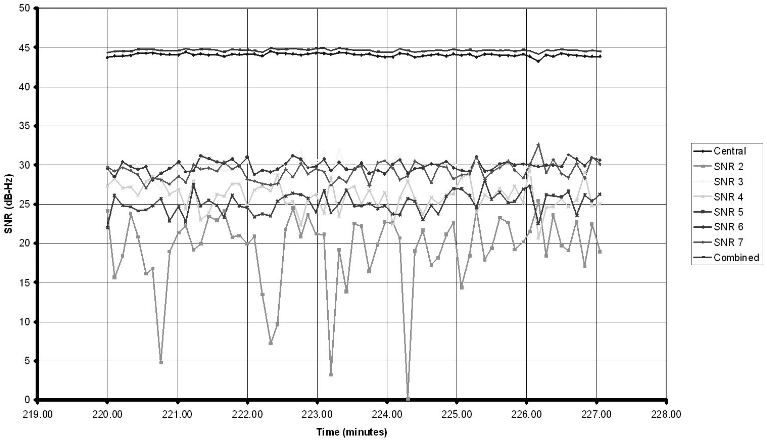

Figure 7.32: Day 195 interpolated least squares track SNR plot.

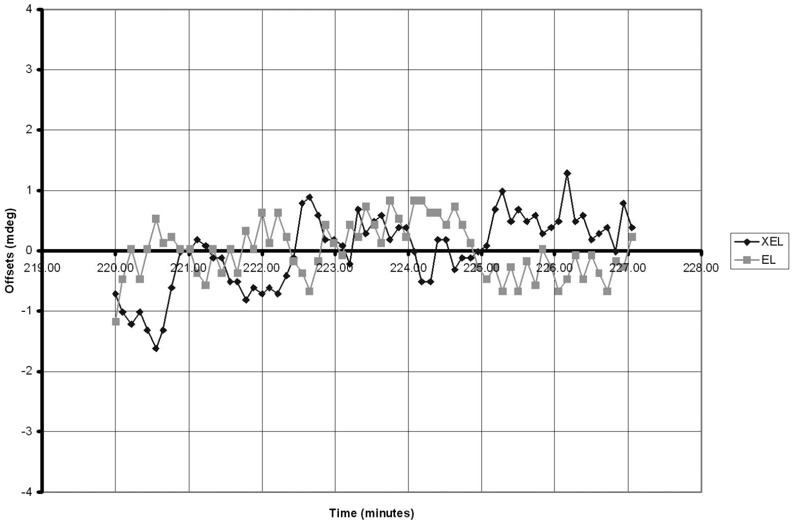

Figure 7.33: Day 195 interpolated least squares track from 49.9 to 51.4 degrees elevation pointing offset plot.

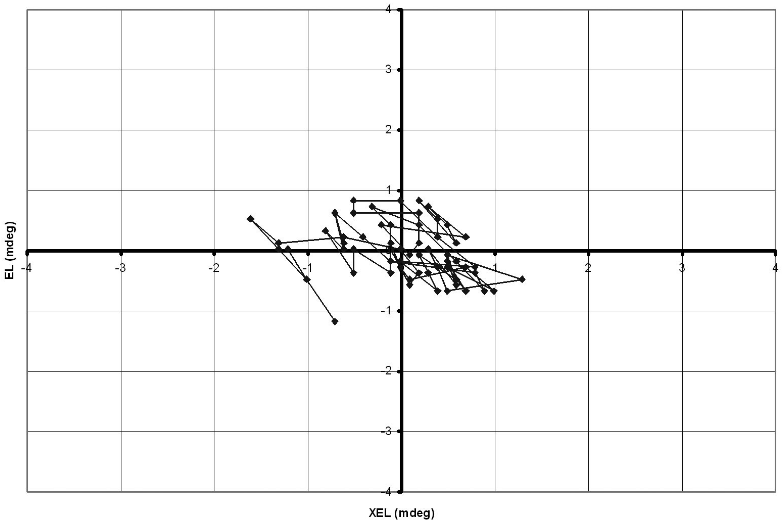

Figure 7.34: Day 195 track with elevation rising from 49.9 to 51.4 degrees. This is a two-dimensional plot of the track in *(XEL,EL)*-space.

7.1.3 Summary and Discussion of DSS-14 results

The results obtained at DSS-14 are summarized in Table 31. Both RBF networks and interpolated least squares algorithms meet and exceed the DSN's pointing accuracy requirements. These results provide an important confirmation of our simulation results, particularly since this antenna is a 70-meter antenna of the same type simulated in Chapter 6. They do not confirm any of our results regarding performance in the presence of focal errors since no such experiments were performed at DSS-14 owing to time constraints.

Table 31: RMS pointing error summary at DSS-14

Day of Year	**Elevation Range (deg)**	**Algorithm**	**RMS Pointing Errors (mdeg) (No trend line removal)**	**RMS Pointing Errors (mdeg) (With trend line removal)**
68	62.7 to 59.2 descending	RBF Network	0.8104	0.3179

68 *part of track above	62.7 to 59.2 descending	RBF Network	0.4219	0.3074
124	40.3 to 39.0 descending	RBF Network	0.4441	0.3775
124	36.0 to 34.6 descending	RBF Network	0.5154	0.4784
171	41.0 to 39.4 descending	RBF Network	0.6428	0.6334
171	39.3 to 37.8 descending	RBF Network	0.6521	0.5798
195	35.3 to 37.0 ascending	RBF Network	0.6242	0.5006
195	39.5 to 41.6 ascending	RBF Network	0.3349	0.3179
195	43.4 to 46.3 ascending	RBF Network	0.4984	0.4157
195	47.9 to 49.9 ascending	RBF Network	0.7007	0.6281
107	41.1 to 39.2 descending	Interpolated Least squares	0.6671	0.6505
167	34.8 to 37.3 ascending	Interpolated Least squares	0.4230	0.4141
195	49.9 to 51.4 ascending	Interpolated Least squares	0.7648	0.6380

Since the performance of the interpolated least squares algorithm is affected by the accuracy and applicability of the underlying lookup table, which is a partial model of the relationship between antenna pointing and focal plane field characteristics, there is reason to believe that the relationship between antenna pointing and focal plane fields as a function of elevation may not change significantly enough to affect these pointing algorithms even when the following parameters are varied:

1. Outside temperature.

2. Day vs. night. Sunlight striking the main reflector causes thermal gradients that can cause thermal expansion to affect main reflector deformation.
3. Ascending in elevation vs. descending in elevation. Although differences in antenna characteristics have been noted during these two distinct phases of operation while tracking spacecraft [47], the applicability of a descending model to an antenna during the ascent phase would suggest that the differences may not be significant for pointing purposes.

The robustness of pointing in the face of these changes is a good result, but it is also unexpected. The following are a few reasons why this result was not expected. In the joint AFCS/DFP experiments of 1999, Vilnrotter and Fort noted a change in array feed combining gain depending on whether the antenna was ascending or descending, and this is indicative of a change in focal plane field characteristics depending on whether the antenna was ascending or descending in pointing elevation [47]. The interpolated nearest neighbor lookup tables, which also served as training sets for the RBF networks, were gathered with the antenna descending on days 52 and 62. Yet tracking operations on days 167 and 195, with the antenna ascending rather than descending, showed good performance of both the interpolated least squares and RBF network algorithms.

Concerning the difference between ascending and descending behavior of the antenna's focal plane field, Imbriale et al. did not find any significant difference in main reflector shape during rising and setting: no evidence of main reflector hysteresis with respect to rising or setting could be detected by theodolite measurements [22]. Yet Imbriale et al. do point out that differences in antenna efficiency have been observed during rising and setting modes [22], and this agrees with differences in combining gain during rising and setting observed by Vilnrotter and Fort [47]. This effect is not fully understood at the time of this writing, but the consistency of pointing algorithm performance from descending to ascending modes appears to agree with the observation that the main reflector's shape apparently does not exhibit hysteresis.

The pointing algorithms maintained consistent behavior even in the presence of significant thermal changes. Although training data and early tracks were carried out under cold night conditions, the day 195 tracks were carried out on a warm summer day. The fact that performance on warmer days remained comparable to cold night performance is a surprise since Imbriale et al. had stated that there is evidence for focal length changes caused by different thermal conditions during day and night operations, with changes potentially being larger than 6-millimeters, possibly causing losses between 5 and 6 db [22]. However, Imbriale et al. have also pointed out that changes in focal length may be partly compensated for by movement of the subreflector due to thermal expansion of the quadripod that attaches the subreflector to the main reflector body, thus reducing such losses [22]. The fact that pointing performance remains almost the same even under daytime conditions even though the reference data were gathered at night may provide limited evidence for the speculation that focal length changes are partly compensated by expansion of the quadripod. It is also possible that the pointing model implied by the lookup table data is robust even in the face of focal errors. Such robustness would seemingly contradict the simulation results of Chapter 6, however, in which it was found that pointing accuracy deteriorated noticeably in the face of previously unseen defocus. This issue has not been fully resolved.

Since the underlying model of the relationship between antenna pointing and the focal plane field as a function of pointing elevation does not appear to change significantly over a long period of time and under varying temperature and sunlight conditions, both interpolated least squares and neural network algorithms that operate with training data gathered under one set of conditions will remain applicable under many other operating conditions as shown by examples from days 167, 171, and 195. This suggests a high degree of robustness of the model itself, which would help to explain the robust performance of RBF networks discussed in Section 7.1.1. Day 124 data illustrate good acquisition performance even with error offsets exceeding those used during training, while day 171 data and day 195 data illustrate the networks' ability to maintain high pointing accuracy and strong SNR recovery performance even under conditions differing from those encountered when the training data were gathered. The set of results from DSS-13 covers focal errors, but caution will have to be exercised in comparing it to the set of 70-meter results and to the simulation results of Chapter 6 due to the fact that the beamwidth is twice as wide at DSS-13 [57].

7.2 Experimental Data from the 34-meter antenna at DSS-13

While a four-millidegree pointing error results in approximately 3-dB of SNR loss on the 70-meter antenna, it would take approximately twice that error to cause the same 3-dB loss at DSS-13 [57]. In this dissertation, it will be assumed that twice the pointing error is considered tolerable for a 34-meter antenna.

Although experiments at the 70-meter antenna at DSS-14 focused exclusively on pointing, experiments at the 34-meter antenna at DSS-13 involved compensation for both pointing and defocus. The DSS-13 experiments used offsets in subreflector position to model defocus errors. In addition, the DSS-13 neural network experiments involved a different type of neural network: the multilayer feedforward network. Our objective was that of evaluating this type of network for DSN use as well. Since the experimental procedure was a lengthy one, which involved three-dimensional raster scans, fewer trials were performed. The resulting data set is smaller than that gathered at DSS-14, but it does provide evidence of the ability of both interpolated least squares and multilayer feedforward network algorithms to correct pointing and focal errors.

7.2.1 Interpolated Least squares Results

The reference table used for interpolated least squares tests was also used as a neural network training set. Since testing on large subreflector position errors (as large as 10-millimeters in some cases) was desired, and since it was necessary to gather raster scan data within a reasonable period of time to avoid excessive elevation changes within a raster scan, the spacing in *Z* was from -6.0-millimeters to +6.0-millimeters in 2.0-millimeter increments. In pointing, the raster scan points ranged from -4.0-millidegres to +4.0-millidegrees in 2.0-millidegree increments. The spacing of 2.0-millimeters in *Z* is, using the conversion factors from Section 4.3.2, approximately 1.06-millimeters in the parabolic deformation parameter *P*.

A set of tests were performed using the interpolated least squares algorithm to correct both subreflector position and antenna pointing. These tests verify the ability of the interpolated least squares algorithm to provide SNR recovery by correcting simultaneous focal and pointing errors and help to verify that this algorithm is a good baseline against which neural networks may be compared. Lookup table data for days 178 and 179 were gathered on day 178.

The first example involves a subreflector-only test, illustrated in Figure 7.35. Here, the antenna was pointed accurately on source, and the resulting SNR was measured. Once the correct pointing offset was known, the subreflector was moved +6.0 mm along the *z*-axis, causing a loss of SNR due to a focal error as illustrated by the second SNR measurement. The interpolated least squares algorithm was then used to bring the subreflector back on focus, and the rise in SNR is visible in the figure. SNR returns to its original value, within measurement error. The antenna was rising in elevation from 54.9 to 55.6 degrees. Since pointing was kept on-source the entire time, no RMS pointing error data were generated. Baseline SNR was 32.7 ± 0.2 dB-Hz in the central channel and 32.8 ± 0.2 dB-Hz in the combined channel, and the SNR after subreflector position correction was, within experimental error, identical to the baseline. In experiments involving pointing, there was usually a slight loss of SNR from the baseline to the tracking phase, although error bars typically overlapped.

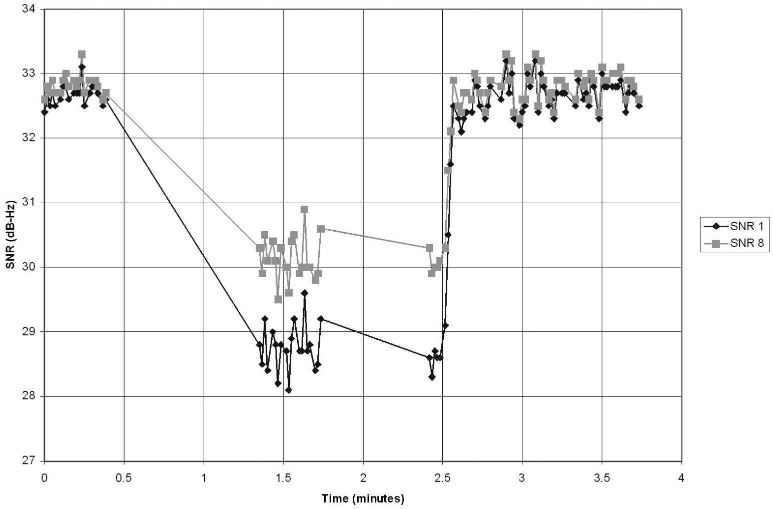

Figure 7.35: Day 178 subreflector-only test.

The second example from day 178, which consists of a combined pointing correction and subreflector correction, is shown in Figure 7.36. The antenna was rising from 74.1 to 74.5 degrees in pointing elevation during this test. The first cluster of points, which completes before the one-minute mark, represents the "baseline" SNR with the antenna pointed and focused properly. The second cluster, just after the two-minute mark, represents an offset of (-4,-4,-8) in *(XEL,EL,Z)* space, with pointing in millidegrees and *Z* in millimeters. Here, the pointing correction is started first, and the third cluster of points crossing the three-minute mark shows a small improvement in SNR. The main improvement occurs when subreflector tracking is turned on, permitting the antenna to return to proper focus and return to the baseline SNR. Here, we have an example of two things: 1. pointing correction and a small SNR gain without focus correction and 2. a return to baseline SNR with focus correction and continued tracking by the pointing algorithm. Steady-state pointing errors were 0.6541-millidegrees RMS. However, since pointing updates were applied only once every 15 seconds due to the relatively slow speed of the least squares algorithm and due to the fact that updates of the antenna are slower at high pointing elevation, the RMS pointing data are not completely reliable. RMS subreflector errors were 1.1929-millimeters. Baseline SNR was 32.6 ± 0.3 dB-Hz for the central channel and 32.8 ± 0.2 dB-Hz for the combined channel. SNR after recovery was 32.4 ± 0.3 dB-Hz for the central channel and 32.6 ± 0.3 dB-Hz for the combined channel.

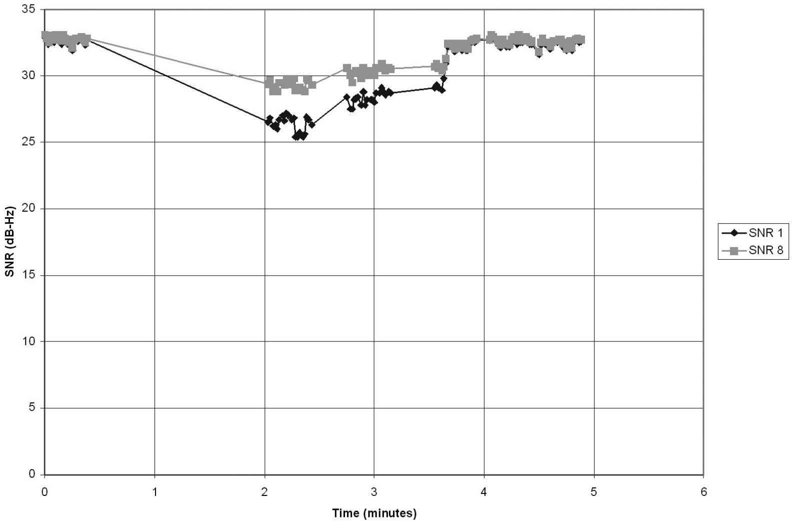

Figure 7.36: Pointing and subreflector correction.

A day 179 example illustrates true simultaneous correction of both a pointing offset and a subreflector offset. Here, the antenna was rising in elevation from 29.8 to

30.5 degrees. As in previous tests carried out at DSS-13, we initially measured SNR with the antenna both in focus and on point. The antenna, in this case, was then sent to point (-4,-4,-10) in *(XEL,EL,Z)*-space, and SNR data were gathered again to establish the severity of the SNR drop. In the third phase, the interpolated least squares algorithm was used to apply simultaneous pointing and focus corrections. This is illustrated in Figure 7.37. As with the previous two examples, SNR was returned to nearly its original value within measurement error. Steady-state RMS pointing error during tracking was 1.0136-millidegrees without trend-line removal and 0.9108-millidegrees with trend-line removal. RMS subreflector position error was 1.0571-millimeters. Baseline SNR was 31.8 ± 0.3 dB-Hz for the central channel and 32.1 ± 0.3 dB-Hz for the combined channel. Recovered SNR was 31.6 ± 0.4 dB-Hz in the central channel and 31.8 ± 0.3 dB-Hz in the combined channel.

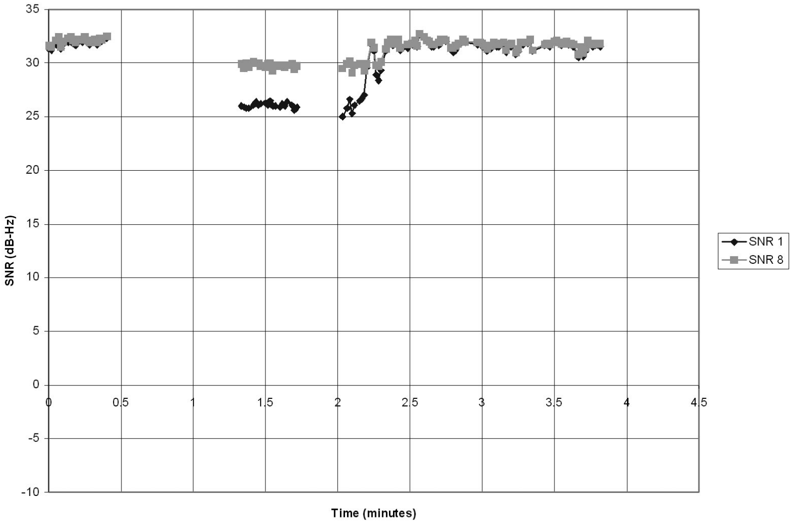

Figure 7.37: Day 179 example 1. Correction of both pointing and focus is shown.

Similar tests are shown in Figure 7.38 (31.8 through 32.7 degrees elevation) and Figure 7.39 (34.7 through 35.9 degrees elevation). The error offsets, in *(XEL,EL,Z)*-space, were (4,-4,-10) in Figure 7.38 and (-4,-4,+6) in Figure 7.39. In each case, the interpolated least squares algorithm was able to return SNR close to the baseline level. RMS pointing errors without trend line removal were 1.1953-millidegrees and 1.4637-millidegrees for Figure 7.38 and Figure 7.39, respectively. With the trend line subtracted, RMS pointing errors are 1.1137-millidegrees and 1.3873-millidegrees, respectively. The RMS subreflector position errors, in steady state, were 0.2281-millimeters and 0.5445-millimeters, respectively. In the test illustrated in Figure 7.38, baseline SNR was 32.0 ± 0.4 dB-Hz in the central channel and 32.2 ± 0.4 dB-Hz in the combined channel. SNR after recovery was 31.5 ± 0.3 dB-Hz in the central channel and 31.7 ± 0.3 dB-Hz in the combined channel. In the test illustrated in Figure 7.39, baseline SNR was 32.0 ± 0.2 dB-Hz in the central channel and 32.3 ± 0.1 dB-Hz in the combined channel. After recovery, measured SNR was 31.3 ± 0.4 dB-Hz in the central channel and 31.5 ± 0.3 dB-Hz in the combined channel.

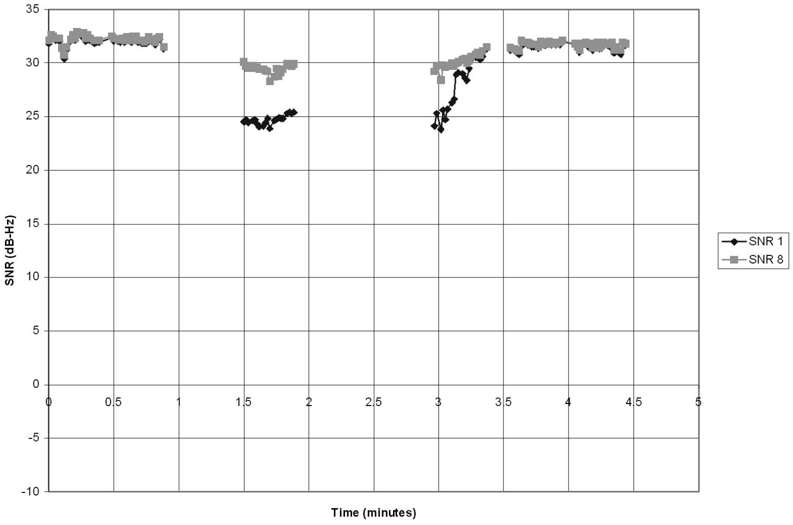

Figure 7.38

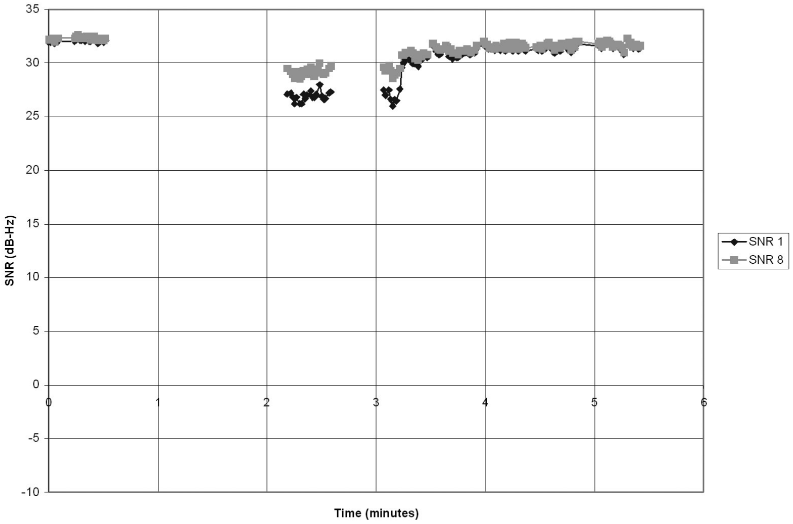

Figure 7.39

7.2.2 Multilayer feedforward network results

On day 180, a series of tests using multilayer feedforward networks were performed. The data used to train three networks, with 30 hidden layer neurons each, were gathered on day 178 and were identical to the data used in the interpolated least squares lookup tables.

The first acquisition and tracking test with neural networks was taken with pointing elevation rising from 29.9 to 30.8 degrees with a starting offset of (-4,-4,-10). As in the interpolated least squares experiments, the “baseline” SNR was found by bringing the antenna to the point (0,0,0) prior to the acquisition and tracking test, and the baseline SNR data were gathered within the first half minute of operations as shown in Figure 7.40. Next, the offset of (-4,-4,-10) was applied, causing an SNR loss shown between 1.5 and 2.0 minutes in the same figure. Acquisition and tracking were performed, leading to the SNR recovery and maintenance of high SNR shown. In the figure, central horn baseline SNR was 31.7 ± 0.3 dB-Hz, and combined channel baseline SNR was 32.0 ± 0.3 dB-Hz at (0,0,0). After acquisition and during tracking, central horn recovered SNR was 31.7 ± 0.3 dB-Hz while combined channel SNR was 31.9 ± 0.3 dB-Hz. The error bar of 0.3 dB-Hz is equal to one standard deviation, and recovered SNR was equal to baseline SNR within experimental error.

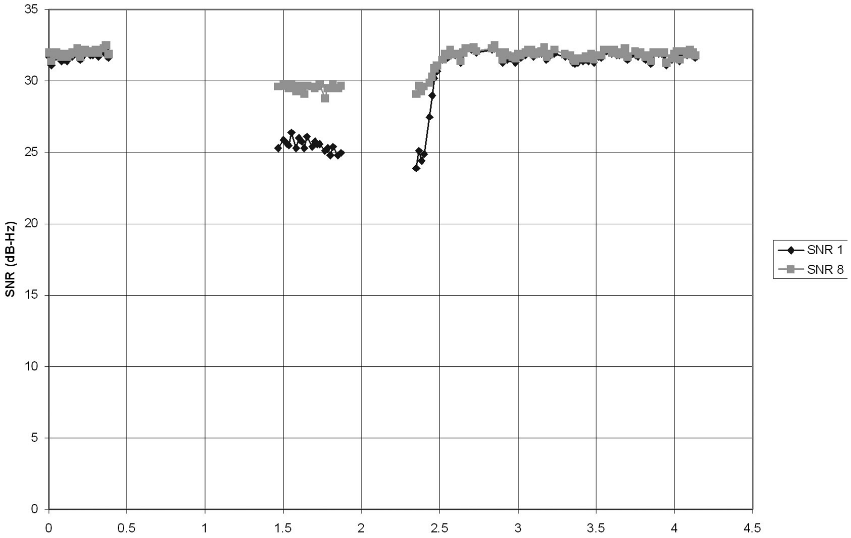

Figure 7.40: SNR during the first multilayer feedforward network track of day 180

Pointing accuracy without trendline removal was 1.0511 millidegrees RMS. With trendline subtraction, pointing accuracy was 0.8909 millidegrees RMS. A plot of pointing during the tracking portion in *(XEL,EL)*-space is shown in Figure 7.41. RMS subreflector position error was 0.4813 millimeters.

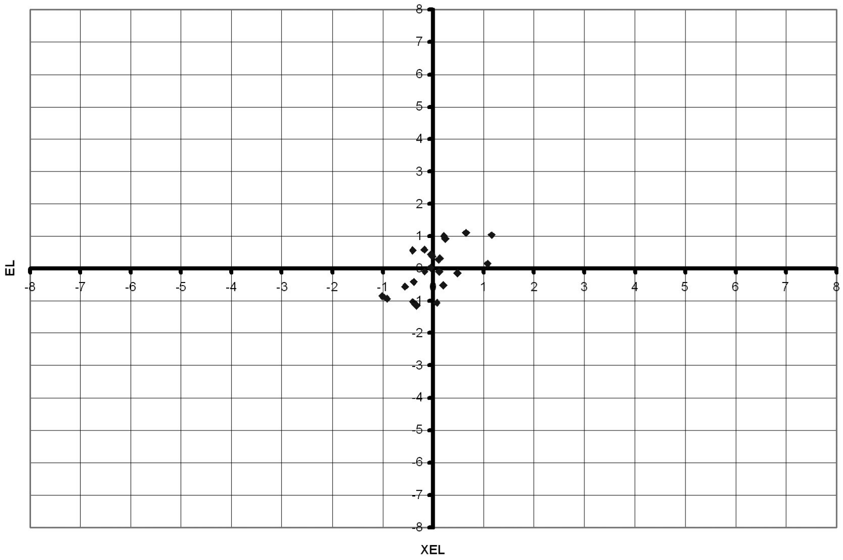

Figure 7.41: Pointing in *(XEL,EL)*-space relative to the source

The next acquisition and tracking test was carried out from a starting offset of (-4,4,-10) with the antenna rising in elevation from 31.1 to 31.8 degrees in elevation. SNR results are shown in Figure 7.42. Baseline SNR was 31.9 ± 0.3 dB-Hz for the central channel and 32.1 ± 0.3 dB-Hz for the combined channel. SNR during tracking after recovery was 31.6 ± 0.3 dB-Hz for the central channel and 31.8 ± 0.3 dB-Hz for the combined channel. Although the error bars for the baseline and recovered SNR values exhibit overlap, indicating they are almost the same within experimental error, there appears to be a slight systematic loss.

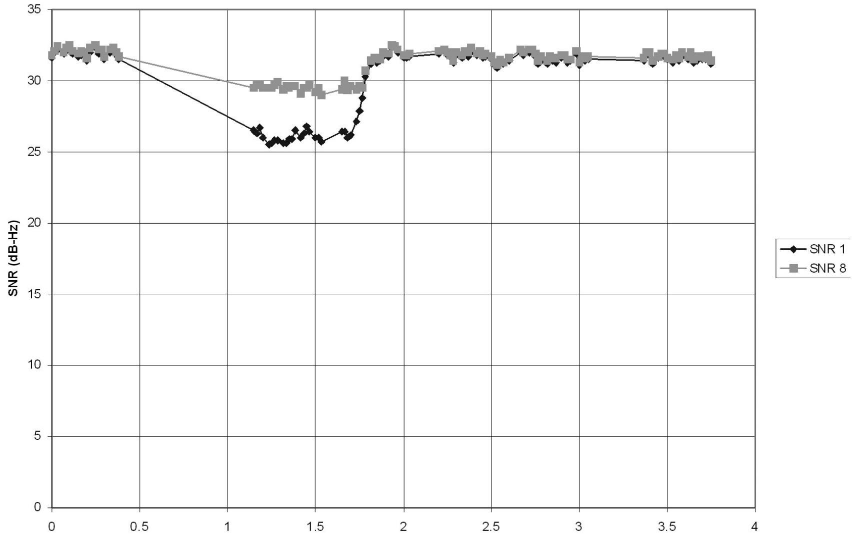

Figure 7.42: SNR during a multilayer feedforward network track of day 180

RMS pointing errors were 0.5420-millidegrees without trend line subtraction and 0.3837-millidegrees with trend line subtraction. RMS subreflector position error was 0.4315-millimeters. A plot of tracking offsets relative to the source is shown.

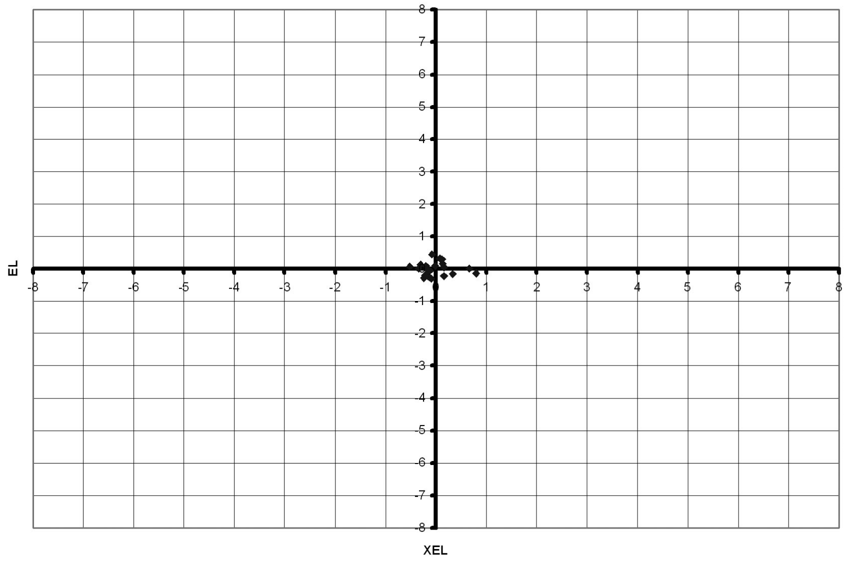

Figure 7.43: Pointing in *(XEL,EL)*-space relative to the source

The next test was performed with the antenna rising in elevation from 32.0 to 32.7 degrees. Baseline SNR was 31.8 ± 0.2 dB-Hz in the central channel and 32.1 ± 0.3 dB-Hz in the combined channel. After recovery, SNR was 31.5 ± 0.4 dB-Hz in the central channel and 31.7 ± 0.3 dB-Hz in the combined channel. The error bars overlap, indicating that these numbers are close within experimental error. Nevertheless, the recovered SNR is slightly lower than the original, signifying that recovery was not perfect. Pointing accuracy was 0.6065-millidegrees RMS without trend line removal and 0.4387-millidegrees with trend-line removal. Subreflector RMS position error was 0.5051-millimeters.

Another test, with a starting offset of (-4,4,6) was conducted with the antenna rising in elevation from 36.5 to 37.1 degrees elevation. Baseline SNR was 32.1 ± 0.2 dB-Hz for the central channel and 32.3 ± 0.2 dB-Hz for the combined channel. After recovery, SNR was 31.5 ± 0.3 dB-Hz for the central channel and

31.7 ± 0.3 dB-Hz for the combined channel. Here, the "one-sigma" (one standard deviation) error bars do not overlap, and not all of the SNR was recovered in this case.

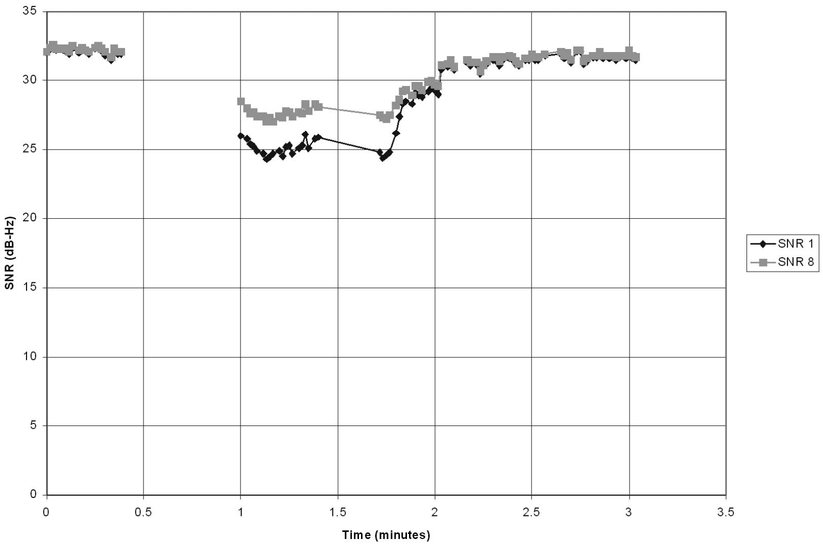

Figure 7.44: SNR during a multilayer feedforward network track of day 180

RMS pointing errors were 0.6993-millidegrees without trend-line subtraction and 0.4801-millidegrees with trend line subtraction. RMS subreflector position error was 0.4187-millimeters.

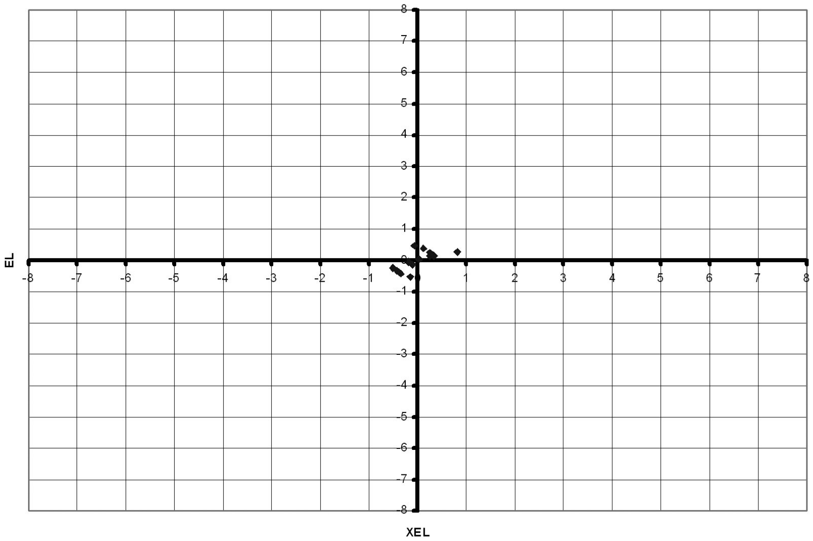

Figure 7.45: Pointing in *(XEL,EL)*-space relative to the source

7.2.3 Summary of experimental results at DSS-13

The following table presents key results from DSS-13.

Table 32: RMS pointing and subreflector *Z*-position errors at DSS-13

Elevation	Algorithm	Pointing RMS (mdeg) No trend removed / trend removed	*Z* RMS (mm)	*P* RMS (mm)	Baseline SNR (central/ combined) (dB-Hz)	Tracking SNR (central/ combined) (dB-Hz)
54.9 to 55.6	Least squares	N/A	0.4351	0.2302	32.7 ± 0.2 / 32.8 ± 0.2	32.7 ± 0.2 / 32.8 ± 0.2
74.1 to 74.5	Least squares	0.6541 N/A	1.1929	0.6312	32.6 ± 0.3 / 32.8 ± 0.2	32.4 ± 0.3 / 32.6 ± 0.3
29.8 to 30.5	Least squares	1.0136 / 0.9108	1.0571	0.5593	31.8 ± 0.3 / 32.1 ± 0.3	31.6 ± 0.4 / 31.8 ± 0.3
31.8 to 32.7	Least squares	1.1953 / 1.1137	0.2281	0.1207	32.0 ± 0.4 / 32.2 ± 0.4	31.5 ± 0.3 / 31.7 ± 0.3
34.7 to 35.9	Least squares	1.4637 / 1.3875	0.5445	0.2881	32.0 ± 0.2 / 32.3 ± 0.1	31.3 ± 0.4 / 31.5 ± 0.3
29.9 to 30.8	Neural Network	1.0511 / 0.8909	0.4813	0.2547	31.7 ± 0.3 / 32.0 ± 0.3	31.7 ± 0.3 / 31.9 ± 0.3
31.1 to 31.8	Neural Network	0.5420 / 0.3837	0.4315	0.2283	31.9 ± 0.3 / 32.1 ± 0.3	31.6 ± 0.3 / 31.8 ± 0.3
32.0 to 32.7	Neural Network	0.6065 / 0.4387	0.5051	0.2672	31.8 ± 0.2 / 32.1 ± 0.3	31.5 ± 0.4 / 31.7 ± 0.3
36.5 to 37.1	Neural Network	0.6993 / 0.4801	0.4187	0.2215	32.1 ± 0.2 / 32.3 ± 0.2	31.5 ± 0.3 / 31.7 ± 0.3

From Table 32, we note that under similar conditions (elevation rising from 29.8 to 37.1 degrees) neural networks appear to yield somewhat lower RMS pointing errors than the interpolated least squares algorithm. In one case, the interpolated least squares algorithm yielded somewhat better focal error performance than did the neural networks, but in most cases neural network performance was either the same or better.

In summary, real-world experiments at both DSS-14 and DSS-13 show that neural networks achieve performance at least as good as that achieved by the interpolated least squares algorithm, in agreement with simulation results. Both interpolated least squares algorithms and neural networks achieve excellent SNR recovery and high pointing accuracy that exceeds DSN requirements, and both are strong candidates for actual implementation.

8 Conclusions and Future Directions

Real world experimental results from Chapter 7 provide the strongest evidence that both interpolated least squares algorithms and neural networks can meet and exceed DSN pointing accuracy requirements and achieve excellent SNR recovery in the face of both pointing errors and focal errors. In agreement with simulation results, neural networks yield performance that is as good as that achieved by interpolated least squares algorithms. Neural networks offer the additional advantages of computational efficiency and, more importantly, adaptivity in the face of changes in the antenna. Experiments show that the algorithms presented in this dissertation can already meet the SNR recovery and tracking accuracy goals of the DSN.

In future research, a maximum likelihood approach could be taken to the development of estimators for pointing and focal errors. Present interpolated least squares algorithms are based on constrained minimization of an objective function, but maximum likelihood architectures based on the Fourier transform and point-sampling approximations could be tested, as was done earlier by Zohar and Vilnrotter [57]. This area of research could lead to the development of an estimator that does not rely on the large lookup tables used in the interpolated least squares algorithms but would be very mathematically intensive. An even more realistic model that avoids the point-sampling approximation and that relies on antenna geometry and physical optics in the place of Fourier transforms may be feasible in principle, but the intensive mathematical development required was not pursued in this dissertation.

Although development of maximum likelihood pointing and focal error estimators represents one possible direction, it is not the only direction for future research. Simulation results suggest that for combined focal error and pointing error correction on 70-meter antennas, the multilayer feedforward network architecture is promising. In all cases, this network had been trained using the resilient backpropagation method [16,17] which relies on the sign of the backpropagated gradient information rather than on both sign and magnitude in performing weight updates. This algorithm has yielded faster, more reliable convergence for this case than traditional backpropagation or backpropagation with momentum [16,17], but other training algorithms may yield better networks.

Some of the neural network design and training issues that represent directions for future research include:

1. The effects of different training methods. This issue was touched upon above but has not been systematically investigated for this problem.
2. The number of hidden-layer units to be used. Although simulation evidence suggests as many as 30, computational complexity may be reduced by using fewer hidden layer units and/or combining functions into a single network with multiple outputs. Time constraints prevented testing of such networks at DSS-13 under focal error estimation conditions, although such networks were tested in simulations.

3. The manner of real-time updating of network weights and biases. Resilient backpropagation had been used for the purpose of updating network weights in Chapter 6, resulting in improved performance or “learning through experience”. While this avoids the need for extra raster scans, there is still a need to perform periodic retraining of the system by briefly (for a few minutes) taking a copy of the neural network offline and generating a retrained copy. This can be done while tracking is going on in real-time. However, the LMS algorithm for adaptive linear filters and for decision-feedback equalizers is used to adapt linear combining weights in real-time without the need to gather blocks of training data [13,18,34], and this could be adapted to real-time retraining of neural networks via backpropagation as well [16-18].

Another promising future research direction involves correction of errors using the Deformable Flat Plate, or DFP [11,21,24]. In this case, a deformable metal plate, which acts as an RF mirror, is placed in the path of RF waves as they travel to the focal plane. Experiments involving the DFP indicate its potential to achieve good SNR gains and to correct for even more errors beyond pointing and defocus. At present, the DFP can be used for correcting systematic deformation errors that are discovered through holographic measurements [38], but research into using it for real-time correction has only begun. Optical errors such as coma, astigmatism, and trifoil distortion also affect received SNR and were considered a part of $d_U\left(r,\theta\right)$, the uncorrectable deformation, in our discussion in Chapter 3. Since the AFCS yields a total of seven complex numbers, which are reduced to six if all outputs are normalized by the central horn as we have done here, it should be possible to detect and correct at least some higher order distortions (at least astigmatism) resulting from main reflector deformation using the DFP. In addition, alternate array geometries for the AFCS with closer horn spacing and/or a greater number of horns may permit even better reconstruction of the focal plane field, allowing even more sophisticated algorithms to be used to reconstruct aperture plane phases and gather even more aperture-plane information [7].

The research presented here is thus a starting point. Algorithms that achieve both DSN pointing accuracy requirements and excellent SNR recovery have been designed, simulated, and demonstrated on actual DSN antennas experimentally. Future developments both along the above-mentioned directions and along many other possible directions will achieve greater SNR recovery performance and better robustness under a wider range of operating conditions, but the DSN's key objectives are achievable with the algorithms developed here.

Appendices

Appendix A: The Orthogonal Least Squares Algorithm for Radial Basis Function Networks

The treatment in this Appendix closely follows the paper by Chen, et al. who developed this algorithm [5]. We begin with a set of definitions.

1. Let $\left\{\vec{\mathbf{x}}(m)\right\}$ be the set of *p*-dimensional input vectors for $m = 1, 2, \ldots, M$. These are the *M* input vectors in the training set. In the OLS algorithm, the set of radial basis centers $\left\{\vec{\mu}_n\right\}$ is a subset of $\left\{\vec{\mathbf{x}}(m)\right\}$. The algorithm seeks to choose a parsimonious set of input vectors as the set of radial basis centers for the network.
2. Corresponding to the *M* input vectors, there are *M* desired outputs given by the vector $\vec{\mathbf{d}} = \left[d(1) \quad d(2) \quad \ldots \quad d(M)\right]^T$
3. Let the set of radial basis centers be denoted by $\left\{\vec{\mu}_n\right\}$, and let $\left\{\vec{\mu}_n\right\} \subseteq \left\{\vec{\mathbf{x}}(m)\right\}$. Let *N* denote the number of radial basis centers. Then $N \le M$.
4. Let $\left\{y_i\right\}$ be the set of *actual* RBF network responses corresponding to the input vector set $\left\{\vec{\mathbf{x}}(m)\right\}$ for $i = 1, 2, \ldots, M$. Define $\vec{\mathbf{y}} = \left[y_1 \quad y_2 \quad \ldots \quad y_M\right]^T$ to be the vector of these responses.

5. Let $\{e_i\}$ be the set of errors defined by $e_i = d_i - y_i$ for $i = 1, 2, \ldots, M$. Define $\vec{\mathbf{e}} = \begin{bmatrix} e_1 & e_2 & \ldots & e_M \end{bmatrix}^T$ to be the vector of errors over the data set.

6. Let $\{w_n\}$ be the set of weights in the linear combiner for $n = 0, 1, \ldots, N$. These weights are used to multiply the outputs of the radial basis units (except for w_0, which acts as a bias term). Define $\vec{\mathbf{w}} = \begin{bmatrix} w_0 & w_1 & \ldots & w_N \end{bmatrix}^T$.

7. Let $z'_{nm} = \exp\left(-\left(b\left\|\vec{\mathbf{x}}(m) - \vec{\mu}_n\right\|\right)^2\right)$. Define $\vec{\mathbf{z}}'_n = \begin{bmatrix} z'_{n1} & z'_{n2} & \ldots & z'_{nM} \end{bmatrix}^T$, a vector that contains all of the *n<u>th</u>* radial basis unit's responses to the vectors in the data set. Also define the $M \times 1$ column vector $\vec{\mathbf{z}}'_0 = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T$.

8. Define $\mathbf{Z}' = \begin{bmatrix} \vec{\mathbf{z}}'_0 & \vec{\mathbf{z}}'_1 & \ldots & \vec{\mathbf{z}}'_N \end{bmatrix}$.

9. Define the squared error: $E = \|\vec{\mathbf{e}}\|^2 = \sum_{m=1}^{M} e_m{}^2$.

10. Let $z_{nm} = \exp\left(-\left(b\left\|\vec{\mathbf{x}}(m) - \vec{\mathbf{x}}(n)\right\|\right)^2\right)$. This is the response of a radial basis unit whose center vector is $\vec{\mu}_n = \vec{\mathbf{x}}(n)$ to the input vector $\vec{\mathbf{x}}(m)$. Then define $\vec{\mathbf{z}}_n = \begin{bmatrix} z_{n1} & z_{n2} & \ldots & z_{nM} \end{bmatrix}^T$. Note that the vectors in $\{\vec{\mathbf{z}}'_n\}$ are ultimately chosen from the set of candidates $\{\vec{\mathbf{z}}_n\}$ in OLS. This is equivalent to saying that the center vectors in $\{\vec{\mu}_n\}$ are ultimately chosen from the input vectors $\{\vec{\mathbf{x}}(m)\}$.

Given the above definitions, we can write:

$$\vec{\mathbf{y}} = \mathbf{Z}'\vec{\mathbf{w}} \tag{A.1.1}$$

We can write the error vector as:

$$\begin{aligned} \vec{\mathbf{e}} &= \vec{\mathbf{d}} - \vec{\mathbf{y}} \\ &= \vec{\mathbf{d}} - \mathbf{Z}'\vec{\mathbf{w}} \\ &= \vec{\mathbf{d}} - \begin{bmatrix} \vec{\mathbf{z}}'_0 & \vec{\mathbf{z}}'_1 & \dots & \vec{\mathbf{z}}'_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} \end{aligned} \tag{A.1.2}$$

Since our goal is that of reducing the squared error E below a given threshold with the minimum number of basis units, we would like to know the contribution of each of the N basis units toward this squared error reduction. We note that the minimum squared error is achieved if and only if $\vec{\mathbf{e}}$ is orthogonal to all of the vectors in $\{\vec{\mathbf{z}}'_i\}$ [5,17,18]. Equivalently, this means $\vec{\mathbf{y}}$ is the projection of $\vec{\mathbf{d}}$ into the subspace spanned by $\{\vec{\mathbf{z}}'_i\}$ [5].

Since the vectors in the set $\{\vec{\mathbf{z}}'_n\}$ are not, in general, orthogonal, we can perform Gram-Schmidt orthogonalization, which is summarized by two equations.

$$\vec{\mathbf{a}}_0 = \vec{\mathbf{z}}'_0 \tag{A.1.3}$$

and

$$\vec{\mathbf{a}}_n = \vec{\mathbf{z}}'_n - \sum_{i=0}^{n-1} \frac{\vec{\mathbf{a}}_i \bullet \vec{\mathbf{z}}'_n}{\|\vec{\mathbf{a}}_i\|^2} \vec{\mathbf{a}}_i \tag{A.1.4}$$

where equation (A.1.4) is used for $n = 1, 2, \dots, N$. Define:

$$\alpha_{i,n} = \frac{\vec{\mathbf{a}}_i \bullet \vec{\mathbf{z}}'_n}{\|\vec{\mathbf{a}}_i\|^2} \tag{A.1.5}$$

and

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & \alpha_{0,1} & \alpha_{0,2} & \dots & \alpha_{0,N} \\ 0 & 1 & \alpha_{1,2} & \dots & \alpha_{1,N} \\ 0 & 0 & 1 & \dots & \alpha_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \tag{A.1.6}$$

we can write:

$$\mathbf{Z}' = \mathbf{A}\boldsymbol{\Psi} \tag{A.1.7}$$

Where $\mathbf{A} = \left[\vec{\mathbf{a}}_0 \quad \vec{\mathbf{a}}_1 \quad \dots \quad \vec{\mathbf{a}}_N\right]$. Since the sets $\left\{\vec{\mathbf{z}}'_n\right\}$ and $\left\{\vec{\mathbf{a}}_i\right\}$ are sets of vectors spanning exactly the same subspace, $\vec{\mathbf{e}}$ must be orthogonal to all vectors contained in $\left\{\vec{\mathbf{a}}_i\right\}$ as well.

Combining equations (A.1.2) and (A.1.7), we have:

$$\begin{aligned} \vec{\mathbf{e}} &= \vec{\mathbf{d}} - \mathbf{A}\boldsymbol{\Psi}\vec{\mathbf{w}} \\ &= \vec{\mathbf{d}} - \mathbf{A}\vec{\mathbf{g}} \end{aligned} \tag{A.1.8}$$

where $\vec{\mathbf{g}} = \boldsymbol{\Psi}\vec{\mathbf{w}}$. Letting g_n be the *n<u>th</u>* component of $\vec{\mathbf{g}}$, we can use equation (A.1.8) and write the following:

$$\vec{\mathbf{d}} = \vec{\mathbf{e}} + \mathbf{A}\vec{\mathbf{g}} \tag{A.1.9}$$

which then gives us:

$$
\begin{aligned}
\left\|\vec{\mathbf{d}}\right\|^2 &= \left\|\vec{\mathbf{e}} + \mathbf{A}\vec{\mathbf{g}}\right\|^2 \\
&= \left(\vec{\mathbf{e}} + \mathbf{A}\vec{\mathbf{g}}\right)^T \left(\vec{\mathbf{e}} + \mathbf{A}\vec{\mathbf{g}}\right) \\
&= \vec{\mathbf{e}}^T\vec{\mathbf{e}} + \vec{\mathbf{g}}^T\mathbf{A}^T\vec{\mathbf{e}} + \vec{\mathbf{e}}^T\mathbf{A}\vec{\mathbf{g}} + \vec{\mathbf{g}}^T\mathbf{A}^T\mathbf{A}\vec{\mathbf{g}} \\
&= \vec{\mathbf{e}}^T\vec{\mathbf{e}} + \vec{\mathbf{g}}^T\mathbf{A}^T\mathbf{A}\vec{\mathbf{g}} \\
&= \sum_{m=1}^{M} e_m^2 + \vec{\mathbf{g}}^T \begin{bmatrix} \left\|\vec{\mathbf{a}}_0\right\|^2 & 0 & \ldots & 0 \\ 0 & \left\|\vec{\mathbf{a}}_1\right\|^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \left\|\vec{\mathbf{a}}_N\right\|^2 \end{bmatrix} \vec{\mathbf{g}} \\
&= \sum_{m=1}^{M} e_m^2 + \sum_{n=0}^{N} g_n^2 \left\|\vec{\mathbf{a}}_n\right\|^2
\end{aligned} \tag{A.1.10}
$$

The move from the third line to the fourth line in equation (A.1.10) is possible because $\vec{\mathbf{e}}$ is orthogonal to all vectors contained in $\left\{\vec{\mathbf{a}}_i\right\}$. The move from the fourth line to the fifth is possible because of the orthogonality of the vectors in $\left\{\vec{\mathbf{a}}_i\right\}$.

At this stage, we note that since $\left\|\vec{\mathbf{d}}\right\|^2$ is a constant, each additional vector in the set $\left\{\vec{\mathbf{a}}_i\right\}$ causes a reduction in the total squared error *E*. If we set *N*, the number of vectors in $\left\{\vec{\mathbf{a}}_i\right\}$, equal to *M*, the number of input data vectors, it becomes possible to reduce *E* to zero. This normally results in an excessively large RBF network and in overfitting of the data [5,17], which must be avoided. Instead, we seek to choose the smallest set $\left\{\vec{\mathbf{a}}_i\right\}$ (equivalently, the smallest set $\left\{\vec{\mathbf{z}}'_n\right\}$) that will bring the squared error *E* sufficiently low to meet the squared error threshold.

The procedure to do this involves the use of modified Gram-Schmidt orthogonalization. In the first step, we try all *M* prospective vectors $\left\{\vec{\mathbf{z}}_m\right\}$, finding the one that will yield the maximum value of $g_1^2 \left\| \vec{\mathbf{a}}_1 \right\|^2$ when we set $\vec{\mathbf{a}}_1 = \vec{\mathbf{z}}_m$. This yields the greatest possible reduction in squared error *E* for the given added vector. This entails adding the corresponding input data set vector as a radial basis center. Instead of simply choosing the first vector for $\vec{\mathbf{a}}_1$ as was done in the standard Gram-Schmidt procedure, we choose the vector $\vec{\mathbf{z}}_m$ such that $g_1^2 \left\| \vec{\mathbf{a}}_1 \right\|^2$, with g_1 defined as in equations (A.1.8) through (A.1.10) for *N=1*, is maximized. This produces the smallest *E* as seen in equation (A.1.10).

On the next step, we continue with the modified Gram-Schmidt procedure. Since we have found $\vec{\mathbf{a}}_1$, we invoke equation (A.1.4) using the remaining vectors of $\left\{ \vec{\mathbf{z}}_m \right\}$ as candidates for $\vec{\mathbf{a}}_2$. For each candidate, we must compute $\sum_{n=1}^{2} g_i^2 \left\| \vec{\mathbf{a}}_i \right\|^2$, and we select the candidate which maximizes $\sum_{n=1}^{2} g_i^2 \left\| \vec{\mathbf{a}}_i \right\|^2$ in order, once again, to minimize *E*. The modified Gram-Schmidt process is repeated until *E* is less than or equal to the desired squared error threshold, at which time the process stops and the set of vectors $\left\{ \vec{\mathbf{a}}_n \right\}$ is complete. The corresponding input data vectors are taken as the centers of the basis functions of the new RBF network, and the optimal weights, computed in the process of obtaining $\vec{\mathbf{g}} = \Psi^{-1} \vec{\mathbf{w}}$ at each step, will be used as the linear combiner weights. The final equations used in the modified Gram-Schmidt iterations are given in the paper by Chen et al. [5] in which this algorithm was first proposed.

Appendix B: Backpropagation Learning

Backpropagation learning is a generalization of the LMS algorithm in that it is also based on the concept of steepest descent. Our treatment of backpropagation very closely follows Hagan, et al. [16] to which we refer the reader.

As in the LMS algorithm, we use a squared-error performance measure, with our objective being that of minimizing the squared error. The network is provided with a set of examples of proper behavior $\{(\mathbf{p}_1,\mathbf{t}_1),(\mathbf{p}_2,\mathbf{t}_2),\ldots,(\mathbf{p}_Q,\mathbf{t}_Q)\}$ where:

$$\mathbf{t}_n = \mathbf{f}(\mathbf{p}_n) \qquad \text{(B.1.1)}$$

for all *n=1,2,…,Q*. The objective is to get the network to approximate the function of equation (B.1.1) as accurately as possible. The mean-squared error is approximated by the instantaneous squared error, which is:

$$E = (\mathbf{t}-\mathbf{a})^T(\mathbf{t}-\mathbf{a}) = \mathbf{e}^T\mathbf{e} \qquad \text{(B.1.2)}$$

where $\mathbf{t}$ is the current desired output corresponding to the current input, $\mathbf{a}$ is the network's output in response to the current input, and $\mathbf{e} = (\mathbf{t}-\mathbf{a})$ is the error vector.

Consider the *(i,j)-th* element of the synaptic weight matrix $\mathbf{W}_m$ in the *m<u>th</u>* layer. The weight update equation for this element is [16]:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha\frac{\partial E}{\partial w_{i,j}^m} \qquad \text{(B.1.3)}$$

where α is the *learning rate parameter*. Similarly, for bias elements, we have [16]:

$$b_i^m(k+1)=b_i^m(k)-\alpha\frac{\partial E}{\partial b_i^m} \tag{B.1.4}$$

The *internal activation* of each neuron in the layer is the input to the non-linearity, which is [16,17]:

$$v_i^m=\sum_{j=1}^{S_{m-1}} w_{i,j}^m a_j^{m-1}+b_i^m \tag{B.1.5}$$

where S_{m-1} is the number of neurons in the previous layer or, equivalently, the length of the column vector of inputs to the current layer *m*.

Computing partial derivatives is the key to performing approximate steepest descent, and the backpropagation procedure operates by computing these derivatives. It is useful to decompose the partial derivatives in equations (B.1.3) and (B.1.4) as follows [16]:

$$\frac{\partial E}{\partial w_{i,j}^m}=\frac{\partial E}{\partial v_i^m}\frac{\partial v_i^m}{\partial w_{i,j}^m} \tag{B.1.6}$$

$$\frac{\partial E}{\partial b_i^m}=\frac{\partial E}{\partial v_i^m}\frac{\partial v_i^m}{\partial b_i^m} \tag{B.1.7}$$

Equations (B.1.6) and (B.1.7) both contain the term $\frac{\partial E}{\partial v_i^m}$. This term is referred to as the *sensitivity* by Hagan et al. [16], and it can be computed by the backpropagation procedure.

Using equation (B.1.5), we obtain:

$$\frac{\partial v_i^m}{\partial w_{i,j}^m}=a_j^{m-1} \tag{B.1.8}$$

and

$$\frac{\partial v_i^m}{\partial b_i^m} = 1 \tag{B.1.9}$$

By combining results from equations (B.1.6), (B.1.7), (B.1.8), and (B.1.9) we obtain:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha a_j^{m-1} \frac{\partial E}{\partial v_i^m} \tag{B.1.10}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial E}{\partial v_i^m} \tag{B.1.11}$$

From equations (B.1.10) and (B.1.11) we see that the primary challenge in computing weight updates lies in computing the sensitivities. The chain rule from calculus provides the foundation of backpropagation.

It will be useful to compute the following Jacobian matrix that relates the internal activations of two adjacent layers in the network[16]:

$$\frac{\partial \mathbf{v}^{m+1}}{\partial \mathbf{v}^m} = \begin{bmatrix} \frac{\partial v_1^{m+1}}{\partial v_1^m} & \frac{\partial v_1^{m+1}}{\partial v_2^m} & \cdots & \frac{\partial v_1^{m+1}}{\partial v_{S_m}^m} \\ \frac{\partial v_2^{m+1}}{\partial v_1^m} & \frac{\partial v_2^{m+1}}{\partial v_2^m} & \cdots & \frac{\partial v_2^{m+1}}{\partial v_{S_m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_{S_{m+1}}^{m+1}}{\partial v_1^m} & \frac{\partial v_{S_{m+1}}^{m+1}}{\partial v_2^m} & \cdots & \frac{\partial v_{S_{m+1}}^{m+1}}{\partial v_{S_m}^m} \end{bmatrix} \tag{B.1.12}$$

Using equation (B.1.5), we can write:

$$\begin{aligned} \frac{\partial v_i^{m+1}}{\partial v_j^m} &= \frac{\partial \left(\sum_{l=1}^{S_m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial v_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial v_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial f_m\left(v_j^m\right)}{\partial v_j^m} \end{aligned} \tag{B.1.13}$$

Using equation (B.1.13) to write each element of the Jacobian matrix of equation (B.1.12) gives us [16]:

$$\frac{\partial \mathbf{v}_{m+1}}{\partial \mathbf{v}_m} = \begin{bmatrix} w_{1,1}^{m+1}\frac{\partial f_m\left(v_1^m\right)}{\partial v_1^m} & w_{1,2}^{m+1}\frac{\partial f_m\left(v_2^m\right)}{\partial v_2^m} & \cdots & w_{1,S_m}^{m+1}\frac{\partial f_m\left(v_{S_m}^m\right)}{\partial v_{S_m}^m} \\ w_{2,1}^{m+1}\frac{\partial f_m\left(v_1^m\right)}{\partial v_1^m} & w_{2,2}^{m+1}\frac{\partial f_m\left(v_2^m\right)}{\partial v_2^m} & \cdots & w_{2,S_m}^{m+1}\frac{\partial f_m\left(v_{S_m}^m\right)}{\partial v_{S_m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S_{m+1},1}^{m+1}\frac{\partial f_m\left(v_1^m\right)}{\partial v_1^m} & w_{S_{m+1},2}^{m+1}\frac{\partial f_m\left(v_2^m\right)}{\partial v_2^m} & \cdots & w_{S_{m+1},S_m}^{m+1}\frac{\partial f_m\left(v_{S_m}^m\right)}{\partial v_{S_m}^m} \end{bmatrix}$$

$$= \begin{bmatrix} w_{1,1}^{m+1} & w_{1,2}^{m+1} & \cdots & w_{1,S_m}^{m+1} \\ w_{2,1}^{m+1} & w_{2,2}^{m+1} & \cdots & w_{2,S_m}^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S_{m+1},1}^{m+1} & w_{S_{m+1},2}^{m+1} & \cdots & w_{S_{m+1},S_m}^{m+1} \end{bmatrix} \begin{bmatrix} \frac{\partial f_m\left(v_1^m\right)}{\partial v_1^m} & 0 & \cdots & 0 \\ 0 & \frac{\partial f_m\left(v_2^m\right)}{\partial v_2^m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial f_m\left(v_{S_m}^m\right)}{\partial v_{S_m}^m} \end{bmatrix} \qquad \text{(B.1.14)}$$

$$= \mathbf{W}_{m+1}\dot{\mathbf{F}}_m\left(\mathbf{v}_m\right)$$

where

$$\dot{\mathbf{F}}_m\left(\mathbf{v}_m\right) = \begin{bmatrix} \frac{\partial f_m\left(v_1^m\right)}{\partial v_1^m} & 0 & \cdots & 0 \\ 0 & \frac{\partial f_m\left(v_2^m\right)}{\partial v_2^m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial f_m\left(v_{S_m}^m\right)}{\partial v_{S_m}^m} \end{bmatrix} \qquad \text{(B.1.15)}$$

Summarizing, we can write:

$$\frac{\partial \mathbf{v}_{m+1}}{\partial \mathbf{v}_m} = \mathbf{W}_{m+1}\dot{\mathbf{F}}_m\left(\mathbf{v}_m\right) \qquad \text{(B.1.16)}$$

By relating the internal activations in adjacent layers, we can now compute sensitivities in each of the layers using the chain rule of calculus. Assume that we already know the sensitivities for layer *(m+1)* given by:

$$\mathbf{s}^{m+1} \triangleq \begin{bmatrix} \dfrac{\partial E}{\partial v_1^{m+1}} \\ \dfrac{\partial E}{\partial v_2^{m+1}} \\ \vdots \\ \dfrac{\partial E}{\partial v_{S_{m+1}}^{m+1}} \end{bmatrix} \tag{B.1.17}$$

By the chain rule, we have [16]:

$$\begin{aligned} \mathbf{s}^{m} &= \begin{bmatrix} \frac{\partial E}{\partial v_1^{m}} \\ \frac{\partial E}{\partial v_2^{m}} \\ \vdots \\ \frac{\partial E}{\partial v_{S_m}^{m}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial v_1^{m+1}}{\partial v_1^{m}} & \frac{\partial v_2^{m+1}}{\partial v_1^{m}} & \cdots & \frac{\partial v_{S_{m+1}}^{m+1}}{\partial v_1^{m}} \\ \frac{\partial v_1^{m+1}}{\partial v_2^{m}} & \frac{\partial v_2^{m+1}}{\partial v_2^{m}} & \cdots & \frac{\partial v_{S_{m+1}}^{m+1}}{\partial v_2^{m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_1^{m+1}}{\partial v_{S_m}^{m}} & \frac{\partial v_2^{m+1}}{\partial v_{S_m}^{m}} & \cdots & \frac{\partial v_{S_{m+1}}^{m+1}}{\partial v_{S_m}^{m}} \end{bmatrix} \begin{bmatrix} \frac{\partial E}{\partial v_1^{m+1}} \\ \frac{\partial E}{\partial v_2^{m+1}} \\ \vdots \\ \frac{\partial E}{\partial v_{S_{m+1}}^{m+1}} \end{bmatrix} \\ &= \left(\frac{\partial \mathbf{v}_i^{m+1}}{\partial \mathbf{v}_i^{m}} \right)^T \mathbf{s}_i^{m+1} \end{aligned} \tag{B.1.18}$$

Equation (B.1.18) shows us how to use knowledge of the sensitivities in one layer to compute the sensitivities in the preceding layer, and this permits sensitivity information to be *backpropagated* in the neural network, leading to the term *backpropagation learning* [16]. Once the sensitivities of the neurons of the output layer have been computed, one can repeatedly use equations (B.1.10), (B.1.11), (B.1.16), and (B.1.18) to update the weights and biases in all layers. It now remains to obtain sensitivities in the output layer, which is the layer of neurons whose outputs constitute the outputs of the network as a whole.

Let the *Mth* layer of the network be the output layer. We start with:

$$a_i = f_M\left(v_i^M\right) \tag{B.1.19}$$

This leads to:

$$\frac{\partial a_i}{\partial v_i^M} = \frac{\partial f_M\left(v_i^M\right)}{\partial v_i^M} \tag{B.1.20}$$

Furthermore, we can write:

$$\begin{aligned}\frac{\partial E}{\partial a_i} &= \frac{\partial}{\partial a_i}\left(\sum_{i=1}^{S_M}\left(t_i - a_i\right)^2\right)\\ &= -2\left(t_i - a_i\right)\end{aligned} \tag{B.1.21}$$

The chain rule gives us:

$$\begin{aligned}\frac{\partial E}{\partial v_i^M} &= \frac{\partial E}{\partial a_i}\frac{\partial a_i}{\partial v_i^M}\\ &= -2\left(t - a_i\right)\frac{\partial f\left(v_i^M\right)}{\partial v_i^M}\end{aligned} \tag{B.1.22}$$

We can write the vector of sensitivities for the output layer by converting equation (B.1.22) to vector form. Doing so yields:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}\left(\mathbf{v}_M\right)\left(\mathbf{t} - \mathbf{a}\right) \tag{B.1.23}$$

Note that for the case of a linear output layer, $\dot{\mathbf{F}}\left(\mathbf{v}_M\right) = \mathbf{I}$, and this gives us $\mathbf{s}^M = -2\left(\mathbf{t} - \mathbf{a}\right)$. Using the output layer sensitivities given by equation (B.1.23), we can compute the sensitivities in all of the hidden layers and compute the appropriate weight updates on each iteration.

Summarizing the backpropagation learning algorithm, we compute the network's output vector $\mathbf{a}$ first. The error vector $\mathbf{e} = (\mathbf{t} - \mathbf{a})$ is then computed, and equation (B.1.23) is used to obtain the sensitivities for the output layer. These sensitivities are backpropagated using equation (B.1.18) to obtain the sensitivities for the hidden layers. All updates of weights and biases are then performed using equations (B.1.10) and (B.1.11). Some intermediate equations must be evaluated in the process of computing the Jacobian used in equation (B.1.18) as shown earlier, but the backpropagation algorithm is an elegant application of the chain rule of calculus to the problem of neural network learning.

Bibliography

[1] Baher, F., "Open- and Closed-Loop Analysis of the 70-Meter Antenna Subreflector Positioner," *TMO Progress Report 42-145*, vol. pp. 1-15, May 15, 2001.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-145/145B.pdf

[2] Bathker, D. A., Veruttipong, W., Otoshi, T. Y., and Cramer, P. W. Jr., "Beam-Waveguide Antenna Performance Predictions with Comparisons to Experimental Results," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, pp. 1274-1285, Jun, 1992.

[3] M. Born and E. Wolf. *Principles of Optics*, Cambridge, United Kingdom: Cambridge University Press, 1999.

[4] R.N. Bracewell . *Two-Dimensional Imaging*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1995.

[5] Chen, S., Cowan, C. F. N., and Grant, P. M., "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 302-309, Mar, 1991.

[6] Cheung, K.-M. and Vilnrotter, V., "Channel Capacity of an Array System for Gaussian Channels With Applications to Combining and Noise Cancellation," *TDA Progress Report 42-124*, vol. pp. 53-62, Feb 15, 1996.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-124/124D.pdf

[7] Cramer, P. W., "Initial Studies of Array Feeds for the 70-Meter Antenna at 32 GHz," *TDA Progress Report 42-104*, vol. pp. 50-67, Feb 15, 1991.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-104/104E.PDF

[8] Cramer, P. W., Imbriale, W. A., and Rengarajan, S. R., "A Method Using Focal Plane Analysis to Determine the Performance of Reflector Antennas," *TDA Progress Report 42-120*, vol. pp. 78-103, Feb 15, 1995.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-120/120I.pdf

[9] Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303-314, 1989.

[10] Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, vol. 2, pp. 183-192, 1989.

[11] Galindo-Israel, V., Rengarajan, S. R., Veruttipong, W., and Imbriale, W. A., "Design of a correcting plate for compensating the main reflector distortions of a dual shaped system," *IEEE Antennas and Propagation Society International Symposium,* Ann Arbor, MI, pp. 246-249.

[12] Gawronski, W. and Gudim, M. A., "Design and Performance of the Monopulse Control System," *TMO Progress Report 42-137*, vol. 42-137, pp. 1-15, May 15, 1999.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-137/137A.pdf

[13] Godara, L. C., "Application of Antenna Arrays to Mobile Communications, Part II: Beam-Forming and Direction-of-Arrival Considerations," *Proceedings of the IEEE*, vol. 85, pp. 1195-1245,

[14] J.W. Goodman. *Introduction to Fourier Optics, 2nd. ed.*, Boston, Massachusetts: McGraw-Hill, 1996.

[15] Gudim, M. A., Gawronski, W., Hurd, W. J., Brown, P. R., and Strain, D. M., "Design and Performance of the Monopulse Pointing System of the DSN 34-Meter Beam-Waveguide Antennas," *TMO Progress Report 42-138*, vol. 42-138, pp. 1-29, Aug 15, 1999.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-138/138H.pdf

[16] M.T. Hagan, H.B. Demuth, and M. Beale. *Neural Network Design*, Boston, MA: PWS Publishng Company, 1996.

[17] S. Haykin. *Neural Networks: A Comprehensive Foundation*, Englewood Cliffs, New Jersey: Macmillan Publishing Company, 1994.

[18] Haykin, S. *Adaptive Filter Theory, 3rd. ed.*, Upper Saddle River, NJ: Prentice Hall, 1996.

[19] E. Hecht. *Optics, 4th. ed.*, Boston, MA: Pearson Addison-Wesley, 2002.

[20] Iijima, B., Fort, D., and Vilnrotter, V., "Correlator Data Analysis for the Array Feed Compensation System," *TDA Progress Report 42-117*, vol. 42-117, pp. 110-118, May 15, 1994.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-117/117k.pdf

[21] Imbriale, W. A., "Distortion Compensation Techniques for Large Reflector Antennas," *2001 IEEE Aerospace Conference Proceedings,* Big Sky, Montana, USA, pp. 2-799-2/805.

[22] Imbriale, W. A., Britcliffe, M. J., and Brenner, M., "Gravity Deformation Measurements of NASA's Deep Space Network 70-Meter Reflector Antennas," *IPN Progress Report 42-147*, vol. 42-147, pp. 1-15, Nov 15, 2001.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-147/147I.pdf

[23] Imbriale, W. A. and Hoppe, D. J., "Computational methods and Theoretical Results for the Ka-Band Array Feed Compensation System-Deformable Flat Plate Experiment at DSS 14," *TMO Progress Report 42-140*, vol. 42-140, pp. 1-23, Feb 15, 2000.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-140/140I.pdf

[24] Imbriale, W. A., Moore, M., Rochblatt, D. J., and Veruttipong, W., "Compensation of gravity induced structural deformations on a beam-waveguide antenna using a deformable mirror," *Antennas and Propagation Society International Symposium,* pp. 1680-1683, 1995.

[25] Jamnejad, V. *Lateral and Longitudinal Feed Displacement Errors in a Cassegrain Dual Reflector System*, (UnPub)
Notes: Work in progress

[26] Lee, P. and Veruttipong, W., "RF Optics Study for DSS-43 Ultracone Implementation," *TDA Progress Report 42-119*, vol. 42-119, pp. 268-281, Nov 15, 1994.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-119/119L.pdf

[27] Lo, V. Y., "Ka-Band Monopulse Antenna-Pointing Systems Analysis and Simulation," *TDA Progress Report 42-124* , vol. 42-124, pp. 104-112, Feb 15, 1996.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-124/124F.pdf

[28] Mukai, R., Arabshahi, P., and Vilnrotter, V., "An Array Feed Radial Basis Function Tracking System for NASA's Deep Space Network Antennas," *IEEE Joint Conference Neural Networks,* Como, Italy, July 2000.

[29] Mukai, R., Vilnrotter, V., and Arabshahi, P., "Computationally Intelligent Array Feed Tracking Algorithms for Large DSN Antennas," *TMO Progress Report 42-141*, vol. 42-141, pp. 1-16, May 15, 2000.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-141/141I.pdf

[30] Mukai, R., Vilnrotter, V., and Arabshahi, P., "Tracking Performance of Adaptive Array Feed Algorithms for 70-Meter DSN Antennas," *TMO Progress Report 42-143*, vol. 42-143, pp. 1-24, Nov 15, 2000.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-143/143N.pdf

[31] Mukai, R., Vilnrotter, V. A., Arabshahi, P., and Jamnejad, V., "Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1149-1162, Sep, 2002.

[32] Otoshi, T. Y., "Noise Temperature and Gain Losses due to Leakage through Deep Space Network Antenna Perforated Panels at and above 32 Gigahertz," *IPN Progress Report 42-151*, vol. 42-151, pp. 1-17, Nov 15, 2002.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-151/151E.pdf

[33] Petersen, D. P. and Middleton, D., "Sampling and Reconstruction of Wave-Number-Limited Functions in N-Dimensional Euclidean Spaces," *Information and Control*, vol. 5, pp. 279-323, 1962.

[34] Proakis, J. G. *Digital Communications, 3rd. ed.*, New York: McGraw-Hill, 1995.

[35] Richter, P. , Franco, M., and Rochblatt, D., "Data Analysis and Results of the Ka-Band Array Feed Compensation System-Deformable Flat Plate Experiment at DSS 14," *TMO Progress Report 42-139*, vol. 42-139, pp. 1-29, Nov 15, 1999.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-139/139H.pdf

[36] Richter, P. H. and Rochblatt, D. J., "Raster-Scan for Calibration of DSN Antennas," *TMOD Technology and Science Program News*, vol. no. 9, pp. 5-6,13-14, Apr, 1998. Jet Propulsion Laboratory. Pasadena, CA.
Notes:
http://tmot.jpl.nasa.gov/Program_Overview_Information/IND_Program_News/ind_program_news.html

[37] Rochblatt, D., Hoppe, D., Imbriale, W., Veruttipong, W., Franco, M., Richter, P., and Sink, T., "Methodology, Analysis, and Experimental Study of the DFP/AFCS Systems for Compensating Gravity Distortions on the 70-Meter Antenna," 1999.
Notes:
http://descansosymposium.jpl.nasa.gov/database/pdf/205_roch.pdf

[38] Rochblatt, D. and Vilnrotter, V., "Demonstration of a Ka-Band Array Feed-Deformable Flat Plate Compensation System," *TMOD Technology and Science Program News*, vol. no. 12, pp. 1-2,16-19, Jun, 2000. JPL.
Notes:
http://tmot.jpl.nasa.gov/Program_Overview_Information/Issue12.pdf

[39] Rochblatt, D. J., "A Microwave Holography Methodology for Diagnostics and Performance Improvement for Large Reflector Antennas," *TDA Progress Report 42-108*, vol. 42-108, pp. 235-252, Feb 15, 1992.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-108/108R.PDF

[40] Rochblatt, D. J., "Holographic measurements of the NASA-JPL Deep Space Network antennas," *1998 IEEE Aerospace Conference,* pp. 441-452, 1998.

[41] Rochblatt, D. J. and Rahmat-Samii, Y., "Effects of measurement errors on microwave antenna holography," *IEEE Transactions on Antennas and Propagation*, vol. 39, pp. 933-942, Jul, 1991.

[42] Rochblatt, D. J. and Seidel, B. L., "Microwave antenna holography," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, pp. 1294-1300, Jun, 1992.

[43] Rochblatt, D. J., Withington, P. M., and Jackson, H. J., "DSS-24 Microwave Holography Measurements," *TDA Progress Report 42-121*, vol. 42-121, pp. 252-270, May 15, 1995.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-121/121A.pdf

[44] Rochblatt, D. J., "Ka-Band DFP/AFCS Demo on the 70-M," 1999.
Notes: http://kilroy.jpl.nasa.gov/anteng/dsn_tech/dfp/dfp_demo.pdf

[45] Rochblatt, D. J., Hoppe, D., Imbriale, W., Franco, M., Richter, P., Withington, P., and Jackson, H., "A Methodology for the Open Loop Calibration of a Deformable Flate Plate on a 70-meter Antenna," *Proceedings of the Millenium Conference on Antennas and Propagation AP2000,* Davos, Switzerland, 2000.

[46] Rodemich, E. R. and Vilnrotter, V. A., "Digital Combining-Weight Estimation for Broadband Sources Using Maximum-Likelihood Estimates," *TDA Progess Report 42-116*, vol. 42-116, pp. 98-109, Feb 15, 1994.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-116/116i.pdf

[47] Vilnrotter, V. and Fort, D., "Demonstration and Evaluation of the Ka-Band Array Feed Compensation System on the 70-Meter Antenna at DSS 14," *TMO Progress Report 42-139*, vol. 42-139, pp. 1-17, Nov 15, 1999.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-139/139J.pdf

[48] Vilnrotter, V., Fort, D., and Iijima, B., "Real-time Array Feed Compensation System Demonstration at JPL," *Multi-Feed Systems for Radio*

Telescopes, Tucson, Arizona, pp. 61-73, 1995.

[49] Vilnrotter, V. and Iijima, B., "Analysis of Array Feed Combining Performance Using Recorded Data," *TDA Progress Report 42-125*, vol. 42-125, pp. 1-13, May 15, 1996.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-125/125C.pdf

[50] Vilnrotter, V. A., "Channel Assignments and Array Gain Bounds for the Ka-Band Array Feed Compensation System," *TDA Progress Report 42-106*, vol. pp. 160-169, Aug 15, 1991.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-106/106M.PDF

[51] Vilnrotter, V. A., "Channel Assignments for Improved Gain in Baseband Array Feed Compensation Systems," *IEEE Transactions on Communications*, vol. 42, pp. 2127-2133, May, 1994.

[52] Vilnrotter, V. A. and Rodemich, E. R., "A Real-Time Signal Combining System for Ka-band Feed Arrays Using Maximum-Likelihood Weight Estimates," *TDA Progress Report 42-100*, vol. 42-100, pp. 81-99, Feb 15, 1990.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-100/100H.PDF

[53] Vilnrotter, V. A. and Rodemich, E. R., "A Digital Combining-Weight Estimation Algorithm for Broadband Sources With the Array Feed Compensation System," *TDA Progress Report 42-116*, vol. 42-116, pp. 86-97, Feb 15, 1994.
Notes: http://tmo.jpl.nasa.gov/tmo/progress_report/42-116/116h.pdf

[54] Vilnrotter, V. A., Rodemich, E. R., and Dolinar, S. J. Jr., "Real-Time Combining of Residual Carrier Array Signals using ML Weight Estimates," *IEEE Transactions on Communications*, vol. 40, pp. 604-615, Mar, 1992.

[55] Vilnrotter, V., "Ka-Band Array Feed Compensation System," *DSN Technology Program News*, vol. pp. 1-4, Jun, 1995.
Notes:
http://tmot.jpl.nasa.gov/Program_Overview_Information/Issue3a.pdf

[56] J.M. Wozencraft and I.M. Jacobs. *Principles of communication engineering*, New York, NY: Wiley, 1965.

[57] Zohar, S. and Vilnrotter, V. A., "Using a Focal-Plane Array to Estimate Antenna Pointing Errors" *JPL Publication 91-45*, vol. 91-45, Oct 25, 1991.