Tracking Performance of Adaptive Array Feed Algorithms for 70-Meter DSN Antennas

R. Mukai,[1] V. Vilnrotter,[1] and P. Arabshahi[1]

This article describes computationally intelligent neural-network and least-squares algorithms for precise pointing of NASA's 70-meter Deep Space Network (DSN) antennas using the seven-channel Ka-band (32-GHz) array feed compensation system (AFCS). These algorithms process normalized data from the seven horns of the array in parallel and thus are more robust and more accurate than inherently serial conventional processing techniques (CONSCAN) currently used by the DSN. A previous article discussed the use of new algorithms for acquisition and estimation of relatively large pointing errors [1] while addressing only briefly the issue of fine tracking near the source. However, neural networks designed specifically for fine-tracking operations yield better fine-pointing performance and significantly lower complexity than those designed for coarse acquisition, and large reductions in complexity may be achieved by using a low-complexity fine-pointing neural network in conjunction with a very simple coarse-acquisition algorithm. In addition to complexity reduction, we also demonstrate the ability to update parameters of the radial basis function (RBF) network in near-real time in response to changes in the antenna, highlighting a useful characteristic of RBF neural networks for antenna-pointing control. The ability to update an RBF network in near-real time without complete restructuring or redesign of the network permits efficient operation even in the presence of frequent changes in the antenna surface.

I. Introduction

A. Tracking in the Presence of Antenna Distortions

An accepted technique for recovering losses due to gravitational deformations, thermal distortion, and wind is by means of a real-time compensation system employing an array of feeds in the focal plane, as described in [1,2]. A seven-element focal-plane array feed compensation system designed to recover gravitational losses on large DSN antennas has been constructed and evaluated at the Goldstone Deep Space Communications Complex (GDSCC) [2]. This system, called the array feed compensation system (AFCS), was developed to demonstrate real-time gravity compensation and closed-loop tracking of spacecraft and radio-source signals at Ka-band frequencies (nominally 32 GHz).

---

[1] Communications Systems and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

In the absence of antenna distortions, a single properly designed receiving horn collects virtually all of the focused signal power [2]. Antenna surface distortions often lead to a shift in the peak of the signal distribution as well as a redistribution of the signal power in the focal plane. This leads to loss of power in the central channel, which can be effectively recovered by the outer horns of an array placed in the focal plane. When the horn signals are multiplied by complex combining weights matched to the instantaneous magnitude and phase of the signal in each channel, the signal-to-noise ratio (SNR) of the combined channel can be improved, approaching that of an undistorted antenna under ideal conditions. Distortions also affect the pointing of the antenna by introducing shifts in the signal peak. Antenna-pointing errors can degrade the received SNR of both single-horn and array receivers, particularly at Ka-band frequencies. Our intent here is to demonstrate that properly designed neural-network or least-squares algorithms effectively remove the time-varying pointing errors and keep the antenna pointed in the direction of maximum SNR even in the presence of significant antenna distortions.

As in [1], it is convenient to represent the received samples as K dimensional vectors, $\mathbf{r}(i) = \mathbf{s}(i) + \mathbf{n}(i)$, where $\mathbf{r}(i) = (r_1(i), r_2(i), \cdots, r_K(i))$ is the vector of samples over a K-element array at time i. In order to reduce the effects of noise, a large number of consecutive received vectors may be averaged, yielding:

$$\mathbf{r}_a(jL) = (r_{a,1}(jL), r_{a,2}(jL), \cdots, r_{a,K}(jL)) = \frac{1}{L} \sum_{i=jL-L+1}^{jL} \mathbf{r}(i) \tag{1}$$

where

$$\mathrm{var}\big(r_{a,k}(jL)\big) = \frac{\sigma^2}{L} \tag{2}$$

and $r_{a,k}$ refers to the kth component of the averaged vector, σ^2 is the variance of the additive white Gaussian noise, and $\mathbf{r}_a(j)$ is the averaged vector output at time jL, where $j = 1, 2, 3, \cdots$.

B. Using AFCS Data for Tracking Purposes

It has been shown [1,4] that data from the AFCS can be used to estimate and to correct pointing errors to achieve maximum SNR at the horns. The instantaneous pointing error vector (XEL, EL) is a two-dimensional error vector. The incremental pointing error in cross-elevation is XEL, and the incremental pointing error in elevation is EL. Both are measured in millidegrees (mdeg).

We seek to compute the mapping from the K-dimensional voltage vector $\mathbf{r}_a$ to the two-dimensional error vector (XEL, EL). This mapping may be represented as

$$\begin{bmatrix} XEL \\ EL \end{bmatrix} = f(\mathbf{r}_a) \tag{3}$$

Residual errors in the voltage vector $\mathbf{r}_a$ due to noise cause errors in the estimate of (XEL, EL) even if $f(\mathbf{r}_a)$ is known exactly. However, $f(\mathbf{r}_a)$ is also affected by the physical structure of the antenna, which is not always precisely known and which changes as the antenna ages or is buffeted by wind. The noisy and time-varying nature of $f(\mathbf{r}_a)$ poses an additional challenge.

Our objective is to develop the processing function $f(\mathbf{r}_a)$. Both interpolated least-squares algorithms and radial basis function (RBF) networks have been shown to provide good approximations to the underlying function, allowing pointing offsets to be accurately estimated using the complex voltage data from the horns.

II. Algorithm Descriptions

Two algorithms, a radial basis function network and a quadratic interpolated least-squares algorithm, were developed to synthesize the function $f(\mathbf{r}_a)$ described by Eq. (3), and theoretical descriptions of both are given in the following subsections.

A. Algorithm 1: Radial Basis Function Network

A radial basis function (RBF) neural network was developed and used to estimate antenna pointing errors [3,5]. As shown in Fig. 1, this radial basis algorithm takes 12 real inputs at a time, which are the real and imaginary parts of the six normalized outer horn voltages. The RBF network consists of two layers: a nonlinear radial basis function layer and a linear combiner layer. Each radial basis unit implements a Gaussian function of the form

$$G(\mathbf{r}_a(jL);\mathbf{c}_i) = \exp\left(-\left(b\|\mathbf{r}_a(jL)-\mathbf{c}_i\|\right)^2\right) = \exp\left(-\left(b\sum_{k=1}^{12}\left(r_a(jL)_k - c_{ik}\right)\right)^2\right) \tag{4}$$

where $\mathbf{r}_a(jL)$ is the 12-element averaged input voltage vector at time jL [previously defined in Eq. (1)], $\mathbf{c}_i$ denotes the ith radial basis center, and $b = 0.8326/\text{spread}$ controls the width of the unit's region of response. The scalar b is defined so that $G(\mathbf{r}_a(jL);\mathbf{c}_i) = 0.5000$ when $\|\mathbf{r}_a(jL) - \mathbf{c}_i\| = \text{spread}$.

Define the matrix $\mathbf{G}$ as [5]

$$\mathbf{G} = \begin{bmatrix} G(\mathbf{r}_a(L);\mathbf{c}_1) & G(\mathbf{r}_a(L);\mathbf{c}_2) & \cdots & G(\mathbf{r}_a(L);\mathbf{c}_M) & 1 \\ G(\mathbf{r}_a(2L);\mathbf{c}_1) & G(\mathbf{r}_a(2L);\mathbf{c}_2) & \cdots & G(\mathbf{r}_a(2L);\mathbf{c}_M) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ G(\mathbf{r}_a(NL);\mathbf{c}_1) & G(\mathbf{r}_a(NL);\mathbf{c}_2) & \cdots & G(\mathbf{r}_a(NL);\mathbf{c}_M) & 1 \end{bmatrix} \tag{5}$$

where M is the number of radial basis units and N is the number of consecutive input voltage vectors applied to the network during the course of operation.

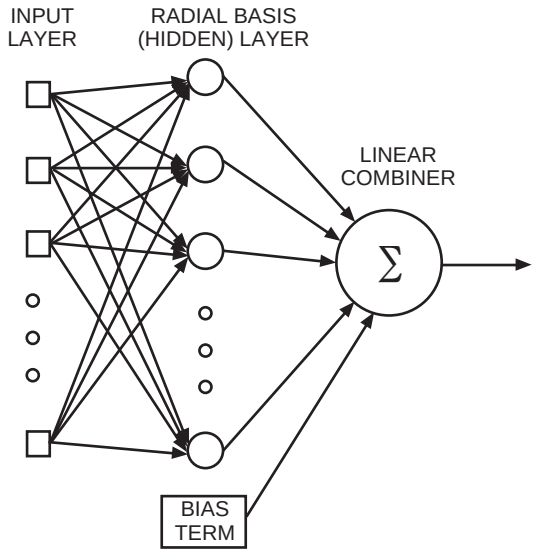

Fig. 1. The RBF neural network.

The column of ones on the right side of $\mathbf{G}$ is necessary since the network contains a bias weight in its linear combiner. A bias weight c_{bias} may simply be thought of as a weight connected to a fixed input of $+1$ rather than to a radial basis unit.

Defining the weight vector as [5]

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \\ c_{\text{bias}} \end{bmatrix} \tag{6}$$

the output of the radial basis function network in response to the N input voltage vectors $\mathbf{r}_a(L)$ through $\mathbf{r}_a(NL)$ as computed by the linear combiner is

$$\mathbf{y} = \begin{bmatrix} y(L) \\ y(2L) \\ \vdots \\ y(NL) \end{bmatrix} = \mathbf{G}\mathbf{w} \tag{7}$$

where $y(jL)$ is the RBF network's response to the jth averaged input vector $\mathbf{r}_a(jL)$ defined in Eq. (1) for $j = 1, 2, 3, \cdots$.

Therefore, each of the radial basis units receives a 12-element input vector and computes the squared Euclidean distance from the input vector to its 12-dimensional basis center vector. The output of the radial basis unit is a number that depends on this squared distance. It is equal to unity when the input vector coincides perfectly with the center of the basis unit; if the input vector does not coincide with the basis unit center, then the basis unit's output will be a positive number less than unity, decreasing to zero in the limit as the squared Euclidean distance between the basis unit center and the input vector goes to infinity. There could be any number of basis units, which are also referred to as hidden layer units, in an RBF network. This number varies depending on the complexity of the function being approximated and may be quite large in certain cases. The number of basis units is determined by the orthogonal least-squares (OLS) algorithm [3] discussed in Subsection II.A.1.

1. The Orthogonal Least-Squares Learning Algorithm. The RBF networks were designed using the computationally efficient orthogonal least-squares algorithm described in [3] and [5]. This algorithm uses the training data points as radial basis function centers, where Gaussian radial basis functions are employed in the single hidden (radial basis) layer and a linear combiner with bias weights is used in the output layer. The optimum weights are obtained as the solution to a least-squares fitting problem, as outlined in [3,5].

Since the complex voltage of the center horn was always normalized to $1 + j0$, it was not necessary to provide this input to the network. This normalization effectively eliminates possible time-dependent variations in the received signal, at the cost of a slight reduction in the total information presented to the algorithms. The network's inputs, therefore, consisted of the real and imaginary components of the six normalized horn voltages from the outer horns, for an input vector dimension of 12. The network was trained to generate values for the incremental EL and XEL offsets corresponding to the inputs.

Differences in the antenna's distortion at different gross elevations led to the training of separate RBF networks for gross elevations of 15, 45, and 75 deg. The experimentally derived basis widths and the

number of basis units required are shown in Table 1. The selection of radial basis widths was guided by distances among voltage vectors in the training set and by experimentation. Since different networks were used for XEL and for EL, there were a total of six networks.

Table 1. Radial basis neural network parameters.

Gross EL, deg	Radial basis spread	Variable estimated	Number of basis units (M)
15	0.6250	XEL	6
15	0.6250	EL	3
45	0.7500	XEL	6
45	0.7500	EL	8
75	0.7500	XEL	13
75	0.7500	EL	15

B. Algorithm 2: Quadratic Interpolated Least Squares

Consider two vector spaces: the 12-dimensional voltage space and the 2-dimensional (XEL, EL) space. An initial estimate for the antenna pointing offset is obtained by finding the voltage vector in the training (reference) set closest to the observed voltage vector. The corresponding vector in (XEL, EL) space is our initial estimate of the pointing offset. Let us call it $(XEL_{\text{initial}}, EL_{\text{initial}})$.

Next, we take the eight closest points to $(XEL_{\text{initial}}, EL_{\text{initial}})$ defined over a rectangular grid in (XEL, EL) space. These points are taken from the training set with a 1-mdeg spaced rectangular training grid as described in Section III. This reference table is also the training set for the RBF network. Once these eight points are taken, it is possible to use one of two methods to obtain (XEL_{est}, EL_{est}). The first method, discussed in [1], was designed to use all eight points and to compute a best-fit quadratic surface. The coordinates (XEL, EL) where the minimum of this surface occurred were taken to be the best estimate.

However, previous research suggests this method offers only a limited advantage over a much simpler quadratic interpolation method [1], where separate one-dimensional quadratic interpolations are used for XEL and EL instead of the more complex two-dimensional surface; it is the simpler method that is used here.

III. Simulation Description

A. Description of Acquisition and Tracking Operations

Spacecraft signals received by the seven-channel AFCS were simulated using surface-distortion data obtained from holography measurements [2]. The computed fields were used to obtain the complex voltages at the output of each horn. The array voltage vector was normalized by the complex voltage of the center horn in order to minimize the effects of external variations in the signal common to all horns.

Two distinct but related problems have been considered: acquisition and tracking. The acquisition problem, which is the focus of [1], involves estimation of antenna-pointing errors over a wide range in (XEL, EL) space. For example, if the antenna's pointing has drifted by 4 mdeg in XEL and -3 mdeg in EL, an acquisition algorithm must be able to estimate the pointing offset $(4,-3)$ accurately and repoint the antenna in the correct direction. The second problem, tracking, is the focus of this article. Tracking

focuses on significantly smaller antenna pointing offsets. After the antenna has been correctly pointed on source by the acquisition algorithm, the tracking algorithm keeps the antenna pointed on source despite slow drifts in antenna pointing. Accordingly, the tracking algorithm must be able to estimate small pointing errors near the center of (XEL, EL) space. For this reason, a fine test grid with 0.25-mdeg spacing was used to test fine-pointing accuracy near the center.

The first algorithm considered here is the RBF network, which offers excellent performance at high SNR and the ability to adapt to changes in the antenna caused by age and by seismic activity. The second algorithm is the quadratic interpolated least-squares algorithm. This algorithm offers the advantage of computational simplicity and robustness over a wide SNR range. Both algorithms were discussed previously in [1] for acquisition purposes, and preliminary results on tracking with the quadratic interpolated least-squares algorithm also were provided in [1].

Tracking simulation results are based on a test grid consisting of points spaced at 0.25 mdeg in both XEL and EL ranging from -2.00 mdeg to 2.00 mdeg in both XEL and EL.

Antenna distortions were simulated at three elevations: 15, 45, and 75 deg, as described in [1]. At each elevation, both training and test sets were created. The training set consisted of points in (XEL, EL) space (with their corresponding 12-dimensional voltage vectors) ranging from -3.00 mdeg to $+3.00$ mdeg in both XEL and EL, spaced uniformly at 1.00-mdeg intervals. This set was used not only as a training set for the RBF network using the OLS [3] procedure but also as a reference table for the quadratic interpolated least-squares algorithm.

The test sets consisted of points in (XEL, EL) space ranging from -2.00 mdeg to $+2.00$ mdeg in both XEL and EL. The points were uniformly spaced at intervals of 0.25 mdeg.

Test sets were prepared for various central horn SNRs, ranging from 10 dB-Hz to 40 dB-Hz in 5 dB-Hz increments. In all cases, independent Gaussian noise was added in all seven horns. The noise variance for a given set was determined using the central horn SNR with the antenna pointed directly on source.

B. Tracking Regions

Although the test sets were defined over a finely spaced grid from -2.00 mdeg to $+2.00$ mdeg in both XEL and EL, it is still useful to evaluate the algorithms' performances over varying ranges in (XEL, EL) space since, for example, one algorithm may yield better performance very close to the on-source direction $(0, 0)$ while another may yield better performance over a greater range of pointing errors. The tracking regions used in the simulation are defined in Table 2.

Error statistics were generated for each of the five tracking regions to illustrate possible performance differences between the RBF neural network and the quadratic interpolated least-squares algorithm.

Table 2. Test regions for performance evaluation.

Region	Region size
1	(0,0)
2	Square. ± 0.50 mdeg in XEL and EL.
3	Square. ± 1.00 mdeg in XEL and EL.
4	Square. ± 1.50 mdeg in XEL and EL.
5	Square. ± 2.00 mdeg in XEL and EL.

IV. Tracking Performance Comparison

Tracking involves estimation and correction of small pointing errors. Tracking algorithms need be optimized only for accurate estimation of errors over a small range; our tests were performed for offsets ranging from −2.00 mdeg to 2.00 mdeg in both XEL and EL.

A. Simulation Results

As previously discussed, five tracking regions were defined for evaluation purposes. The first region was really the point $(0, 0)$ in (XEL, EL) space, and the mean error and error standard deviations of the estimates were determined when the antenna was simulated pointing perfectly on source in the presence of noise. The next region is a square centered at $(0, 0)$, and the remaining regions are progressively larger squares, as defined in Table 2.

1. Neural Network Performance. Figures 2 through 4 show the desired target points in (XEL, EL) space as blue "x" symbols and the estimates computed by the neural network as red dots. For each offset, shown as a blue "x," 100 independent estimates from the neural network were used to obtain a scatter plot of the neural network's estimates of the applied antenna offsets. These red dots form clusters near the blue "x" symbols, and the size and the center of the cluster gives a rough indication of the accuracy of neural network performance.

Each scatter plot was taken at an SNR of 40 dB-Hz with 10-s integration, illustrating performance over region 3 as defined in Table 2. It is interesting to observe that performance is better at low and high elevations, where there is significant antenna distortion, than at 45-deg elevation, where distortion is minimal. This is attributed to the fact that more signal power is present in the outer horns when there is distortion, providing more information to the neural network for a better estimate of pointing error. However, we also observe some noticeable but insignificant bias at 75-deg elevation, evidenced by the fact that the clusters of estimates (red dots) are not perfectly centered on the true offsets (blue "x" symbols). A summary of RBF network performance in terms of mean error and error standard deviation at 40 dB-Hz SNR with 10-s integration is provided in Table 3.

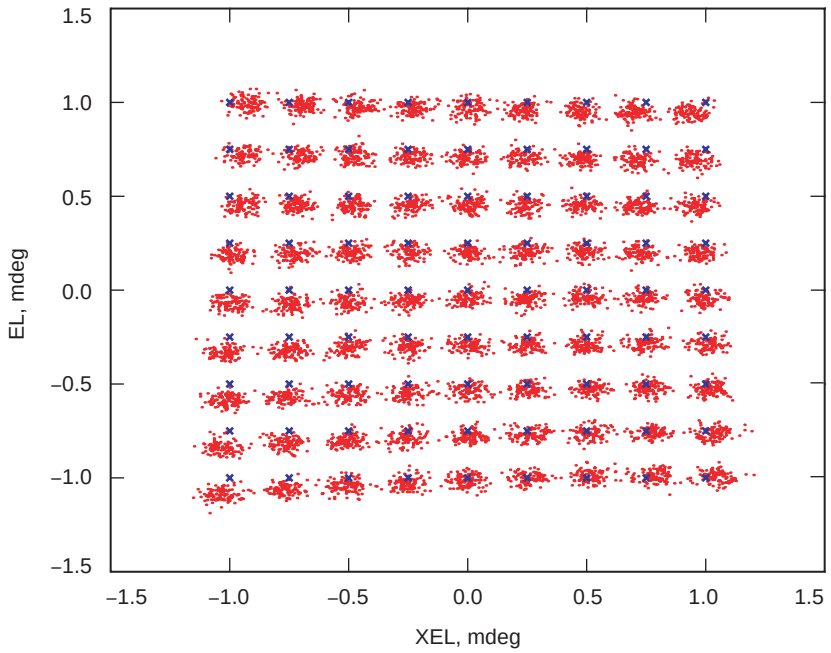

Fig. 2. RBF region 3: 15 deg, 40 dB-Hz, 10-s integration.

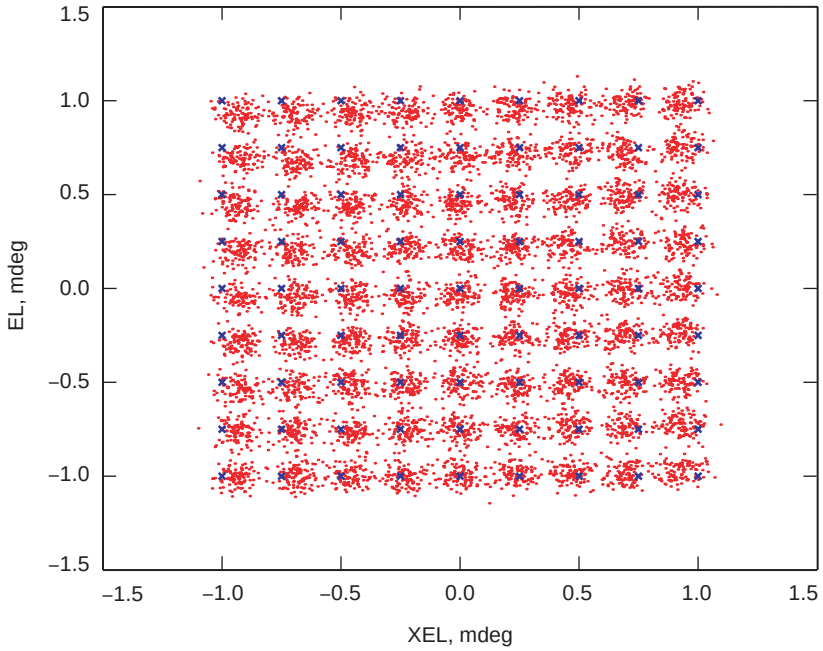

Fig. 3. RBF region 3: 45 deg, 40 dB-Hz, 10-s integration.

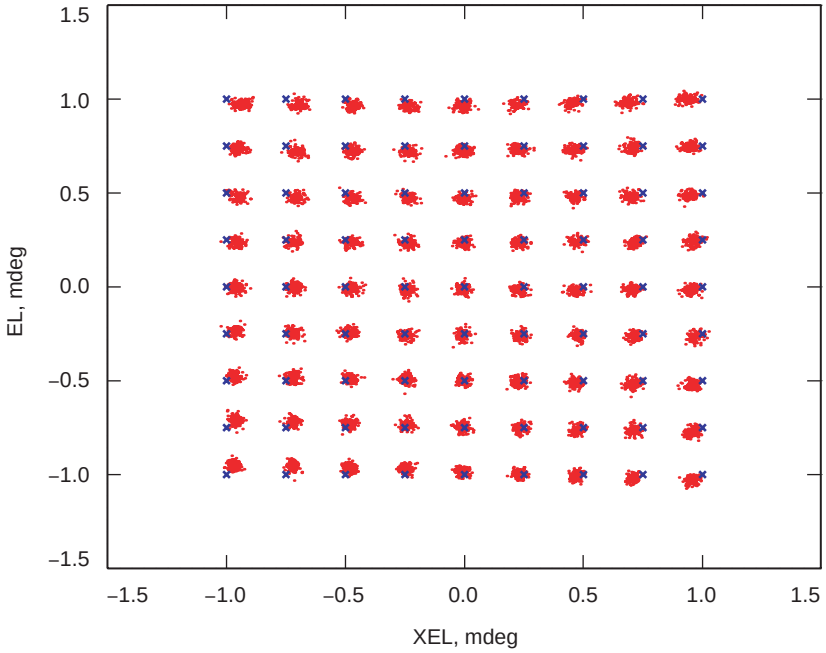

Fig. 4. RBF region 3: 75 deg, 40 dB-Hz, 10-s integration.

2. Least-Squares Performance. We also generated scatter plots for the quadratic interpolated least-squares (LS) algorithm in Figs. 5 through 7. As with the RBF network, performance is better at low elevation than at 45-deg elevation. This is attributed to the greater signal level in the outer horns due to distortion at low elevation. At high elevation, estimator bias starts to become significant, as shown in Fig. 7. The rms error and bias of the least-squares algorithm, averaged over the entire grid, are shown in Table 4.

Table 3. Radial network performance at 40 dB-Hz with 10-s integration.

Elevation, deg	Variable	Mean error, mdeg	Error standard deviation, mdeg
15	XEL	−0.0005	0.0491
15	EL	0.0434	0.0350
45	XEL	0.0009	0.0696
45	EL	0.0231	0.0523
75	XEL	0.0052	0.0380
75	EL	0.0090	0.0258

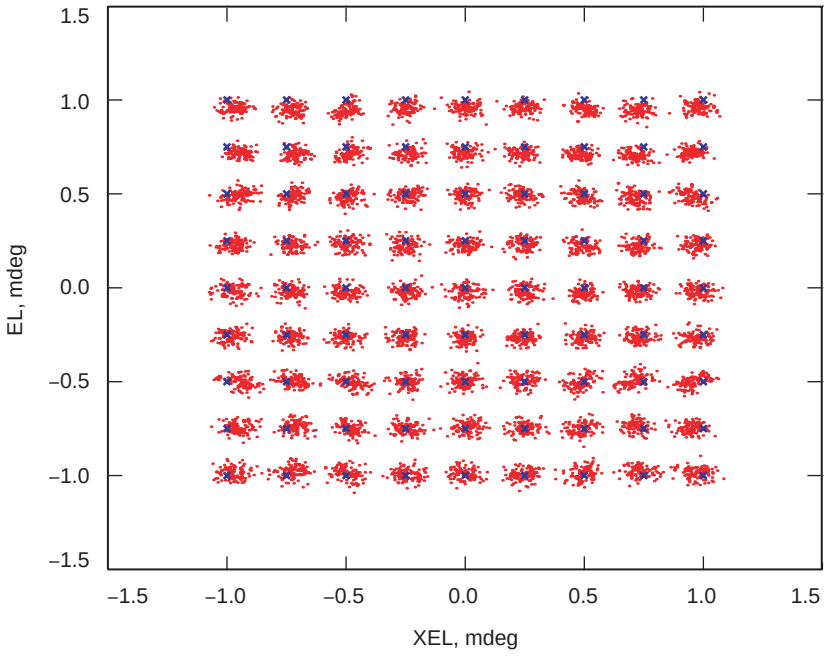

Fig. 5. LS region 3: 15 deg, 40 dB-Hz, 10-s integration.

B. Performance Comparison

The least-squares algorithm evaluated here implicitly assumes a quadratic error surface since it uses quadratic interpolation. This assumption appears to be very good at both 15- and 45-deg gross elevation. It can be seen from Figs. 2 through 7 that the radial basis function network yields similar performance at those two elevations, demonstrating both algorithms' ability to provide accurate tracking under high SNR conditions.

At 75-deg elevation, the antenna surface suffers significant distortion. The spatial distribution of power in the focal plane is greatly affected by these distortions, and a significant amount of power appears in the outer horns. Furthermore, the assumption of a quadratic error surface appears to hold only for a very small neighborhood in this region, and a breakdown of the quadratic assumption is evident as we move away from (0,0) in antenna offset space, as shown in Fig. 7. The RBF network is better able to deal with severe antenna distortion, as illustrated in Fig. 4.

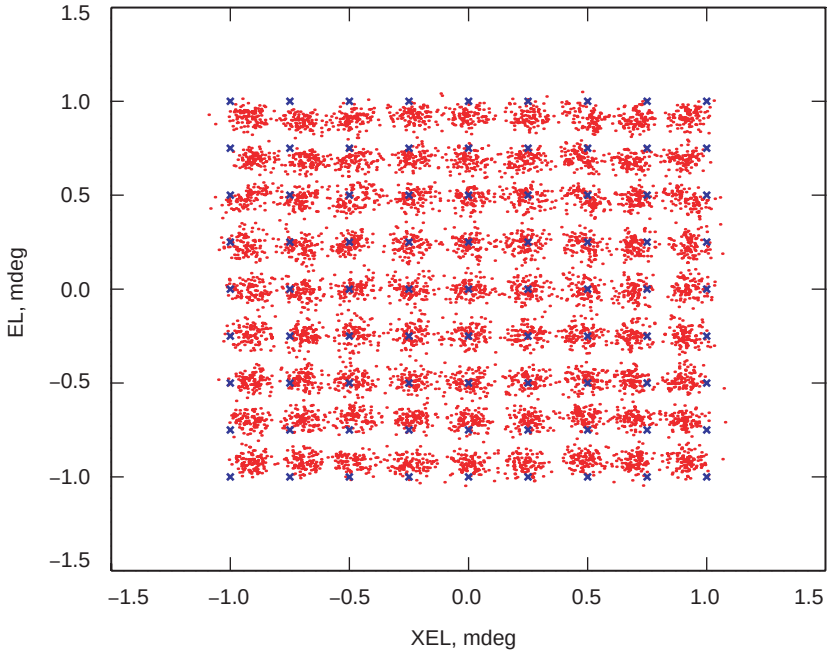

Fig. 6. LS region 3: 45 deg, 40 dB-Hz, 10-s integration.

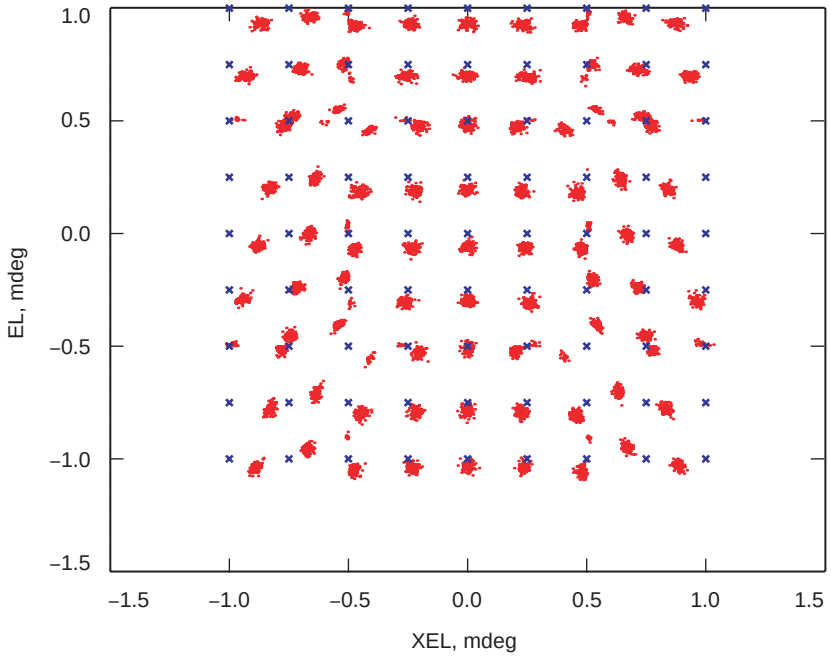

Fig. 7. LS region 3: 75 deg, 40 dB-Hz, 10-s integration.

Under the high SNR conditions shown in Figs. 4 and 7, the clusters are very small for both algorithms at 75-deg antenna elevation. The errors in the least-squares algorithm at this elevation are mostly *systematic* with very little random error, indicating breakdown of the quadratic surface assumption at high elevations when the antenna's main reflector is severely distorted.

Table 4. Quadratic interpolated least-squares performance at 40 dB-Hz with 10-s integration.

Elevation, deg	Variable	Mean error, mdeg	Error standard deviation, mdeg
15	XEL	0.0008	0.0437
15	EL	0.0154	0.0330
45	XEL	0.0001	0.0685
45	EL	0.0034	0.0633
75	XEL	0.0000	0.0819
75	EL	0.0268	0.0422

It is useful to measure both the mean errors and the error standard deviations as a function of SNR over the tracking regions for different integration times. Mean errors over a given region tend to signify systematic errors in the algorithms, while error standard deviations provide a measure of the randomness of the errors. Figures 8 through 19 contain the plots for region 3, defined over ± 1.00 mdeg in antenna offset space, and are representative of results achieved for all five regions. The plots in Figs. 8 through 13 show the mean errors obtained with both the interpolated least-squares and RBF network algorithms. The label "1 s" means we are simulating 1-s integration and generating second-by-second pointing offset estimates. However, averaging over an interval of 10 s rather than 1 s reduces the effects of noise. For this reason, 10-s integration times were also included in the error plots. Error standard deviations are illustrated in Figs. 14 through 19. In all cases, errors in millidegrees are plotted against SNR in dB-Hz.

Figures 18 and 19 show a "knee" in the curve of error standard deviation versus SNR for the RBF network with 1-s integration. As noise standard deviation, σ, begins to exceed the spread of the radial basis units, the basis outputs will begin to decline toward zero. This results in the output of the radial basis network approaching the value of the bias term in the network's linear combiner as σ increases. Since the RBF network's outputs tend to vary less under these conditions, the error variance will not grow very rapidly as SNR declines and will even drop toward zero as SNR becomes very low. The output of the network will eventually be equal to the bias term, leaving only a systematic error as opposed to a random error for extremely low SNR. The SNR of 10 dB-Hz with 1-s integration is not severe enough to produce such saturation, but error standard deviation is still somewhat lower than expected.

It should be noted that RBF networks generally exhibit higher mean errors at low SNRs than the least-squares algorithm does. At medium-to-high SNRs, the RBF network's performance does not differ significantly in terms of mean error. Furthermore, the error standard deviations are comparable, indicating the RBF network comes close to the performance of the quadratic interpolated least-squares algorithm without significant performance differences in random error.

For 75-deg elevation, the RBF network exhibits an advantage in terms of overall performance, mostly due to its ability to better handle nonquadratic error surfaces that are characteristic of severely distorted antenna surfaces.

Overall, both algorithms exhibit similar performance for tracking at elevations of 15 and 45 deg, while RBF networks perform better than the quadratic interpolated least-squares algorithm at very high elevations corresponding to severe antenna distortion.

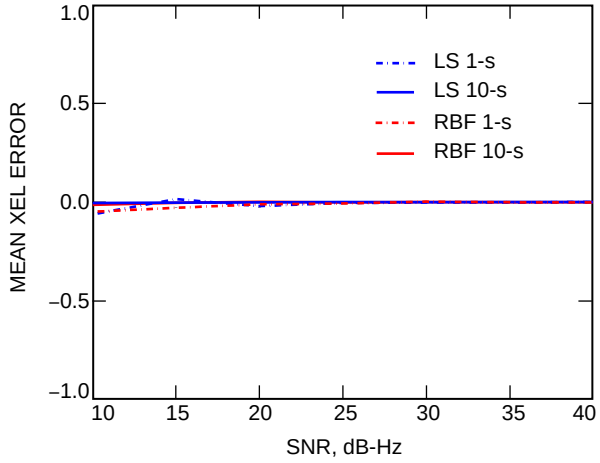

Fig. 8. Mean XEL errors at 15 deg.

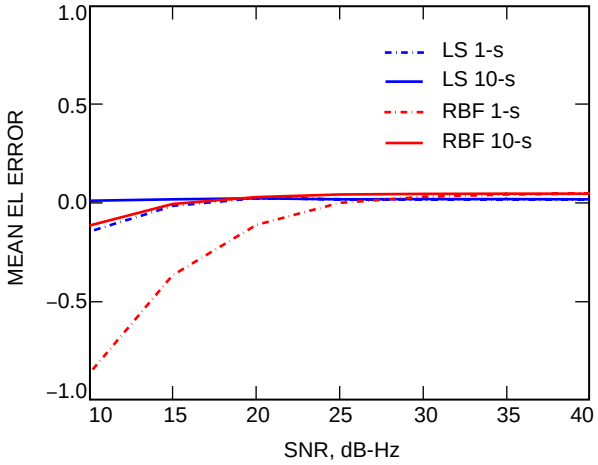

Fig. 9. Mean EL errors at 15 deg.

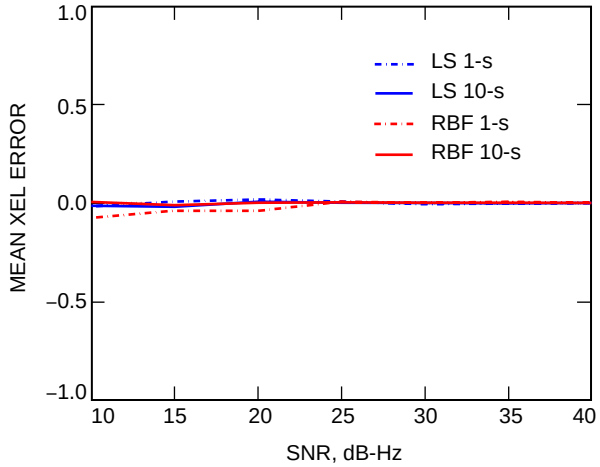

Fig. 10. Mean XEL errors at 45 deg.

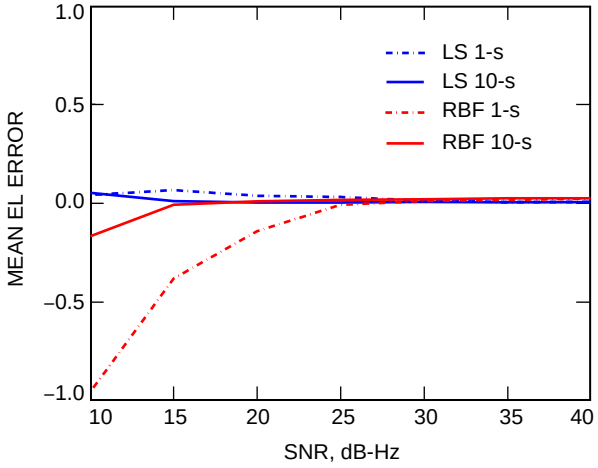

Fig. 11. Mean EL errors at 45 deg.

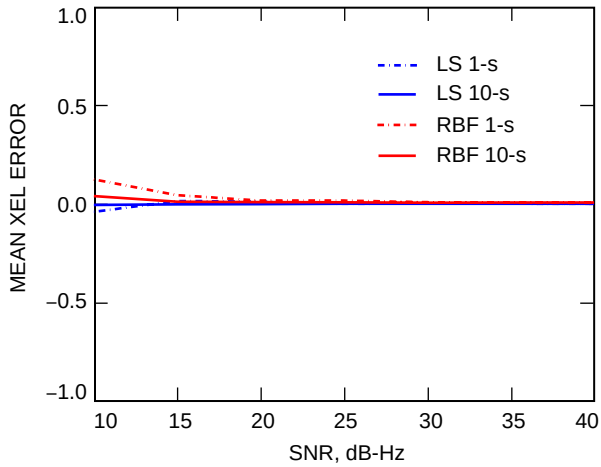

Fig. 12. Mean XEL errors at 75 deg.

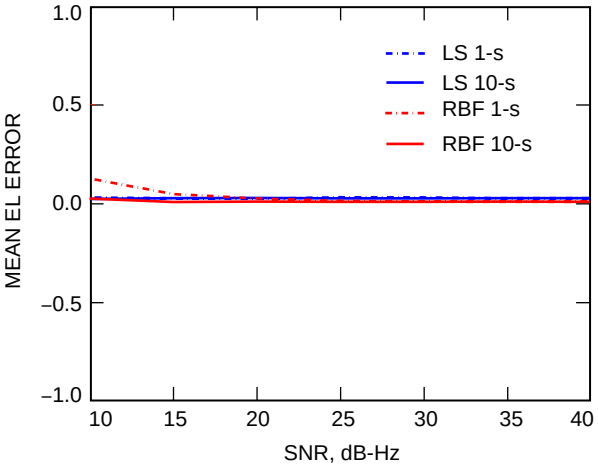

Fig. 13. Mean EL errors at 75 deg.

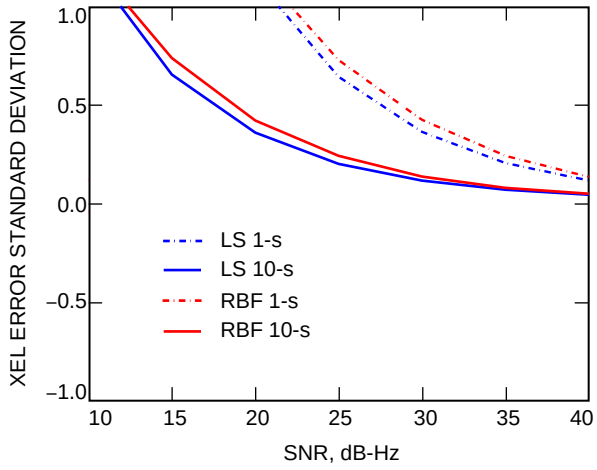

Fig. 14. XEL error standard deviations at 15 deg.

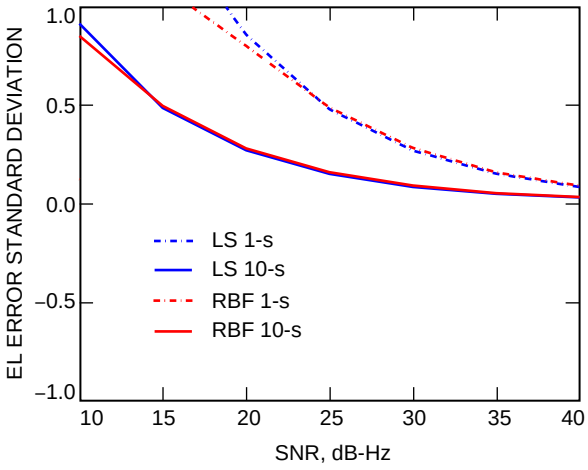

Fig. 15. XL error standard deviations at 15 deg.

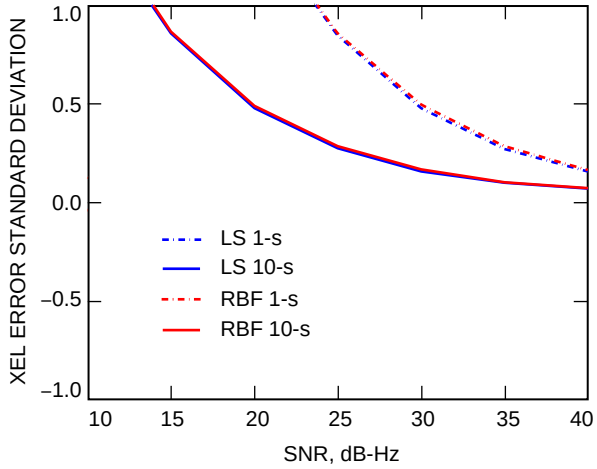

Fig. 16. XEL error standard deviations at 45 deg.

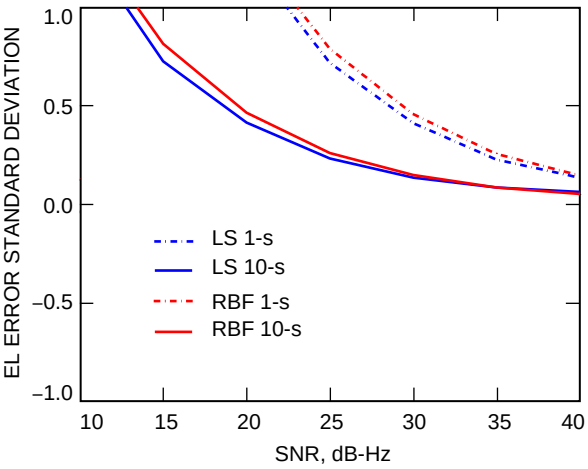

Fig. 17. XL error standard deviations at 45 deg.

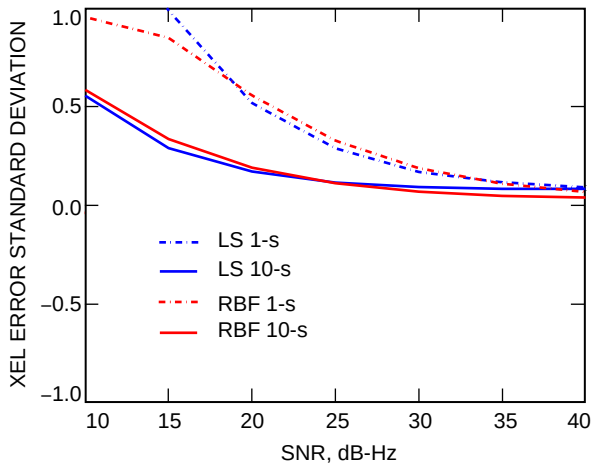

Fig. 18. XEL error standard deviations at 75 deg.

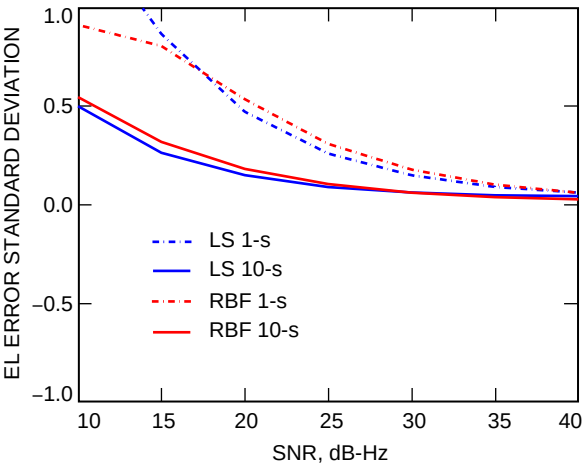

Fig. 19. XL error standard deviations at 75 deg.

We observe that both algorithms achieve the nominal 0.8-mdeg total pointing accuracy at SNRs above approximately 20 dB-Hz at all elevations when 10-s integration is used. This integration time is compatible with the practical update rate of the antenna and represents outstanding weak-signal performance. Even with 1-s integration, both algorithms meet the 0.8-mdeg requirement above 30 dB-Hz, although pointing corrections generally cannot be applied to the antenna at such a high rate.

V. Adaptive Tracking

A. Long-Term Adaptation

The foregoing analysis assumes a stationary antenna model that does not change. In the real world, aging, thermal gradients, and other factors lead to changes in the antenna structure that will, in turn, change the mapping from the 12-dimensional voltage space to the 2-dimensional (XEL, EL) space. This necessitates the use of adaptive tracking algorithms.

In the case of a static antenna model, the RBF network and the interpolated least-squares algorithm both do a very good job of estimating small pointing offsets characteristic of tracking operations. However, RBF networks also offer the advantage of adaptivity. As stated in Subsection II.A, there is a linear combiner at the output stage. One possible adaptation strategy is to leave the radial basis units fixed while only adapting the linear combiner, using either matrix pseudo-inverses or the least-mean-square (LMS) algorithm [5].

The adaptive algorithms described here provide a significant advantage over other adaptation techniques. Haykin [5] does give an algorithm for adapting all of the parameters of an RBF network, but this algorithm does not always converge easily. By contrast, for a *fixed* radial basis layer, it is possible to achieve a global minimum of the mean-squared error as a function of the linear combiner weights [5]. Additionally, the adaptive algorithms below are designed to avoid the need to generate an entire new radial basis function network using the OLS [3] procedure every time the antenna changes, resulting in computational savings and the ability to adjust the RBF network in near-real time.

B. The LMS Algorithm

Since the linear combiner in the radial basis function network may be thought of as a linear adaptive filter, the LMS algorithm may be applied to adjust its weights and its bias term [5].

Let $\mathbf{x}(n)$ be the input training vector at time n, and let $G(\mathbf{x}(n); \mathbf{c}_i)$ denote the output of the ith radial basis unit at time n. Let $w_i(n)$ denote the corresponding linear combiner weight at time n (the weights are updated on each iteration of the LMS algorithm). Let $c_{\text{bias}}(n)$ denote the bias term. The linear combiner calculates [5]

$$y(n) = c_{\text{bias}}(n) + \sum_{i=1}^{N} w_i(n) G(\mathbf{x}(n); \mathbf{c}_i) \tag{8}$$

Updates of the weights are performed by first computing the error [5]:

$$e(n) = d(n) - y(n) \tag{9}$$

where $d(n)$ is the desired output and $y(n)$ is the actual output. The new weights are given by [5]

$$w_i(n+1) = w_i(n) + \eta e(n) G(\mathbf{x}(n); \mathbf{c}_i) \tag{10}$$

To update the bias term, we regard it as a weight attached to a fixed input of +1. Its update is given by

$$c_{\text{bias}}(n+1) = c_{\text{bias}}(n) + \eta e(n) \tag{11}$$

Using Eqs. (10) and (11), we can update the linear combining weights of the radial basis function network.

C. Matrix Pseudo-Inverse

While the LMS algorithm is a powerful procedure, it is a stochastic approximation to true gradient descent. For this reason, convergence is generally somewhat noisy [5]. It is possible to compute the optimal set of linear coefficients for a given training set that will yield the lowest possible mean-squared error (MSE).

Let $\mathbf{r}_j$ denote the jth training set vector, and let $\mathbf{c}_i$ denote the ith radial basis center. The matrix $\mathbf{G}$ defined by Eq. (5) has a pseudo-inverse given by [5]

$$\mathbf{G}^+ = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T \tag{12}$$

The set of linear combining weights that minimize the MSE over the training data is [5]

$$\mathbf{w}_{\text{new}} = \mathbf{G}^+\mathbf{d} \tag{13}$$

where $\mathbf{d}$ is the set of desired outputs at times 1 through N defined by

$$d = \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix} \tag{14}$$

Equation (13) always computes the best possible set of weight updates with respect to the mean-squared error criterion over the set of training vectors. The LMS algorithm, by contrast, tends to converge to the optimal weights using stochastic gradient descent, but it does not always reach the best weights.

D. Simulating Changes in the Antenna Structure

Changes in the antenna structure were simulated by altering the mapping from voltage space to (XEL, EL) space when generating the training sets for the neural networks. Let the mapping from voltage space to offset space be given by Eq. (3), and create a distorted mapping by changing the (XEL, EL) values corresponding to the voltage vectors using, for example, Eqs. (14) and (15):

$$XEL_{\text{old}} = \frac{3}{4}XEL_{\text{new}} + \frac{1}{2}\text{ mdeg} \tag{15}$$

$$EL_{\text{old}} = \frac{5}{4}EL_{\text{new}} + \frac{1}{2}\text{ mdeg} \tag{16}$$

One may think of XEL_{old} and EL_{old} as being the correct offsets corresponding to given voltage vectors for the antenna at time T_1. Aging, seismic events, and other factors may cause the same voltage vectors

to map to XEL_{new} and EL_{new} at time T_2, where we have assumed T_1 and T_2 to be far apart since these are *long-term* changes. The RBF networks were trained on the old data in an effort to generate a set of distorted mappings. We can say that these training data were correct at time T_1 in the distant past but changes in the antenna have yielded a new mapping at time T_2. We seek to update the RBF network parameters so the networks will accurately determine offsets under the new mapping.

In Figs. 20 through 22 (40 dB-Hz and 10-s integration times), we see that the red clusters corresponding to the antenna offsets from the neural networks are very far from the blue "x" symbols denoting the new values. This illustrates the performance of the networks trained on old data, representing, for example, a pointing bias and scaling due to aging. Our objective is to retrain the networks to remove the pointing bias by adapting their linear combiners..

For the 75-deg elevation case, Fig. 23 shows the mean errors resulting from training on the old set. Figure 24 shows the error standard deviations at 75 deg.

E. LMS Results

Using an updated set of training data, the RBF networks were retrained using the LMS algorithm. The resulting networks, with a 10-s integration and an SNR of 40 dB-Hz, yield the performance illustrated in Figs. 25 through 27, showing that the systematic pointing biases have been removed. The mean error and error standard deviation are shown as functions of SNR for the 75-deg case in Figs. 28 and 29.

It can be seen that after LMS retraining the network achieves significantly improved performance, although it does not achieve the same performance as a network trained on the new data using the OLS algorithm. While the pointing biases have been removed, we see from Fig. 26 that the scaling distortion has not been fully removed. This is due to several factors. The number of radial basis units, along with their centers and their widths, remains fixed. It is likely that the optimal set of centers for the old data is not the optimal set of centers for the new data. Furthermore, LMS is a stochastic gradient descent

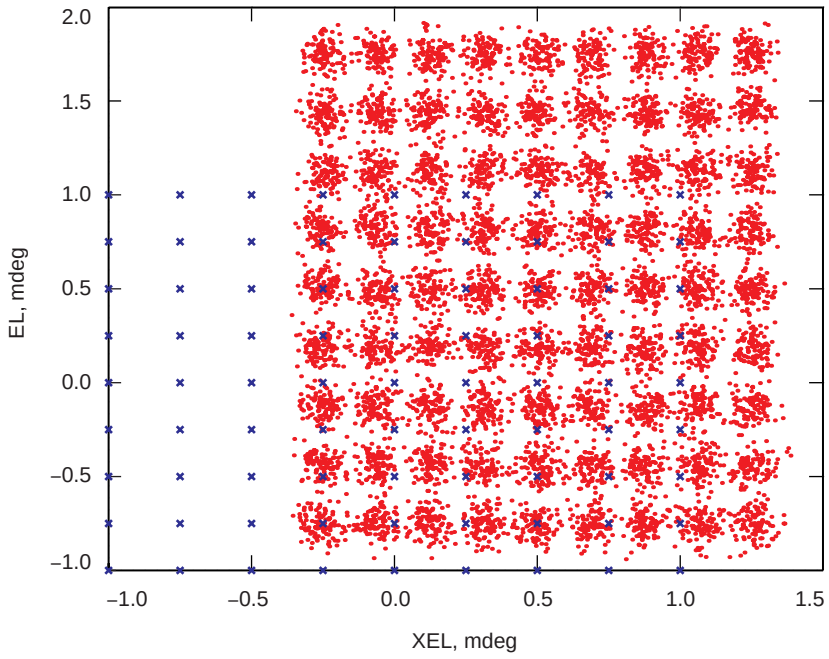

Fig. 20. Performance of the network trained on old data at 15 deg.

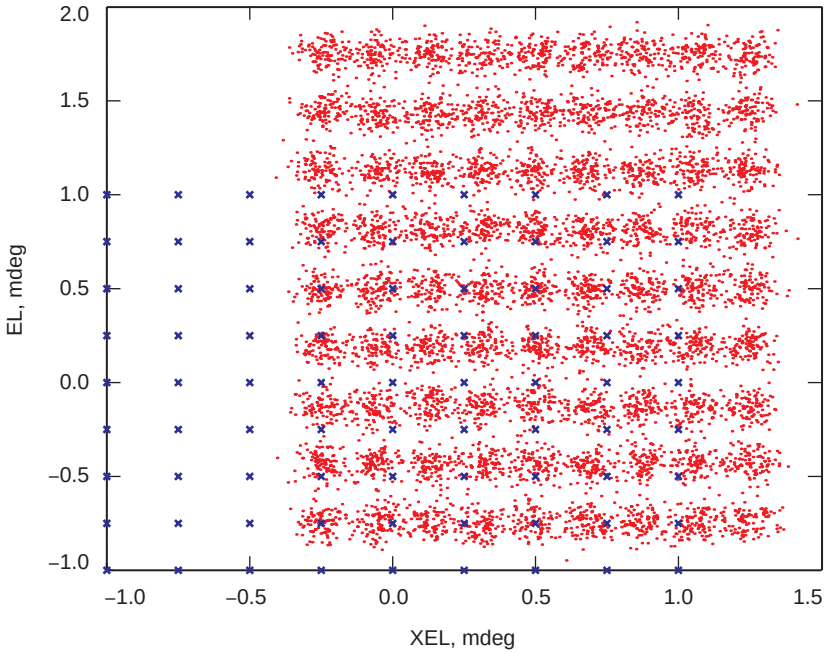

Fig. 21. Performance of the network trained on old data at 45 deg.

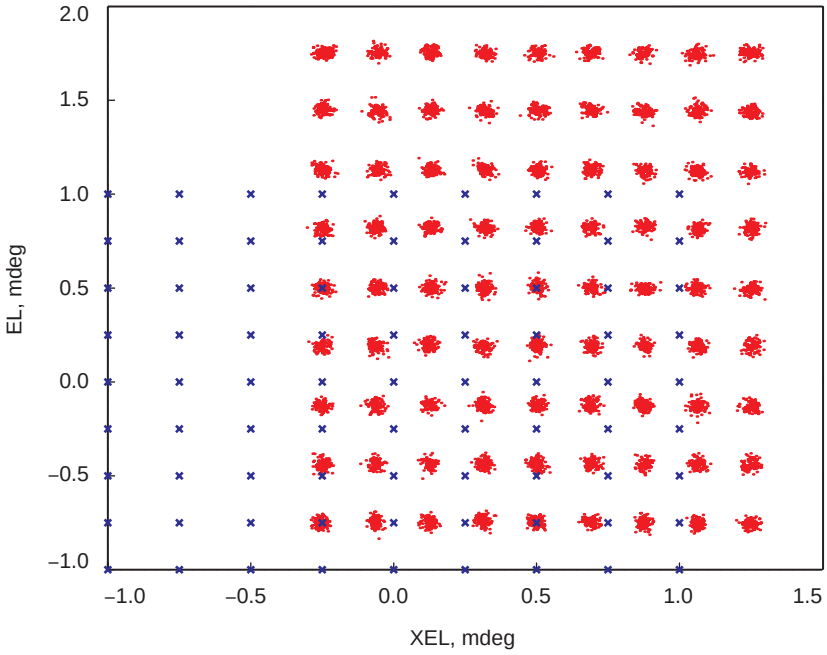

Fig. 22. Performance of the network trained on old data at 75 deg.

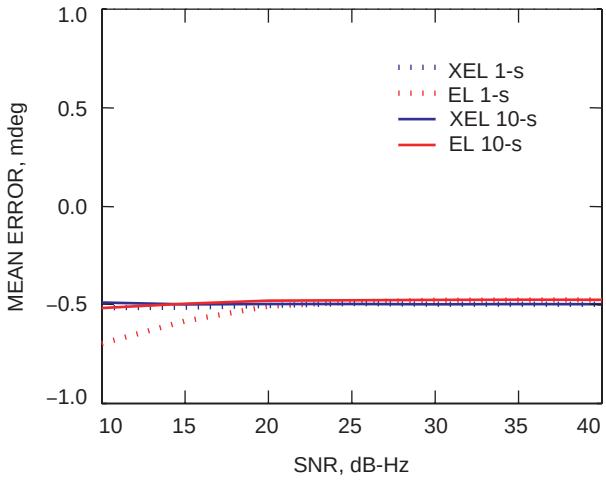

Fig. 23. Mean error resulting from training on old data at 75 deg.

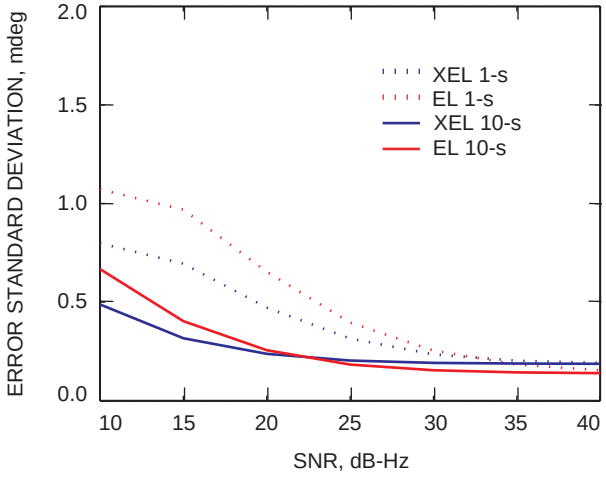

Fig. 24. Error standard deviation resulting from training on old data at 75 deg.

procedure that does not implement true, noiseless gradient descent. There is, therefore, some noise in the convergence process [5]. In Subsection V.F, we will see better error performance since the linear combiner weights are indeed optimal for the given radial basis centers and spreads.

F. Pseudo-Inverse Optimization

As shown in Figs. 30 through 34, the pseudo-inverse method yields superior training performance, with most of the error caused by the old training set well corrected after the retraining procedure is complete. The LMS method also yields acceptable performance, but LMS is a stochastic gradient descent procedure [5]. It is, therefore, only an approximation to true gradient descent, and even true gradient descent takes many iterations to converge asymptotically to the optimal weight vector $\mathbf{w}$ [5]. By contrast, the pseudo-inverse method of recalculating the linear combiner weights calculates a set of weights that yields the lowest possible MSE for a fixed radial basis layer [5].

If near-real-time updating is desired, it may be advisable to use the LMS algorithm, although storing and averaging data to form new training sets to update the linear combiner weights using the pseudo-inverse method would likely yield better performance. While it is possible to use the OLS method to

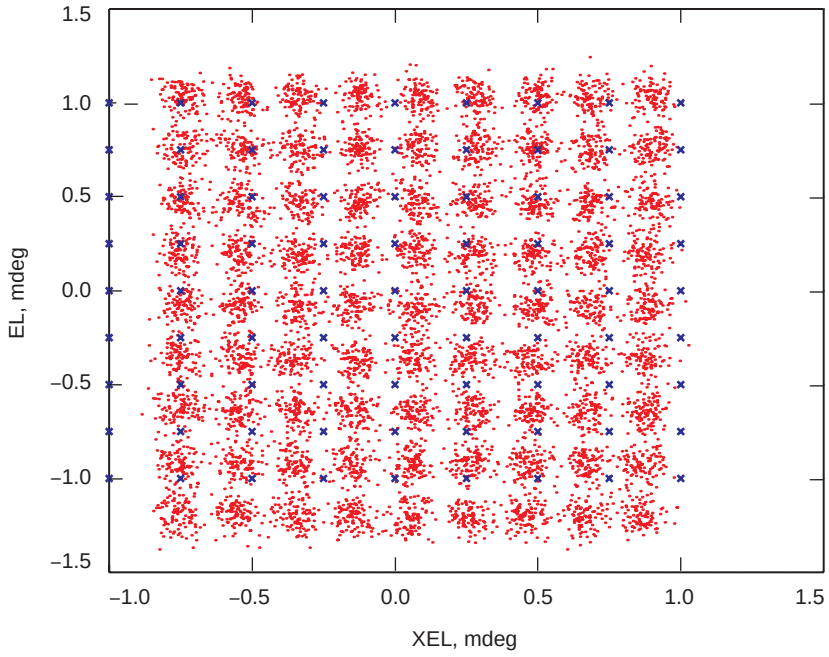

Fig. 25. Performance of the neural network retrained with the LMS algorithm at 15 deg.

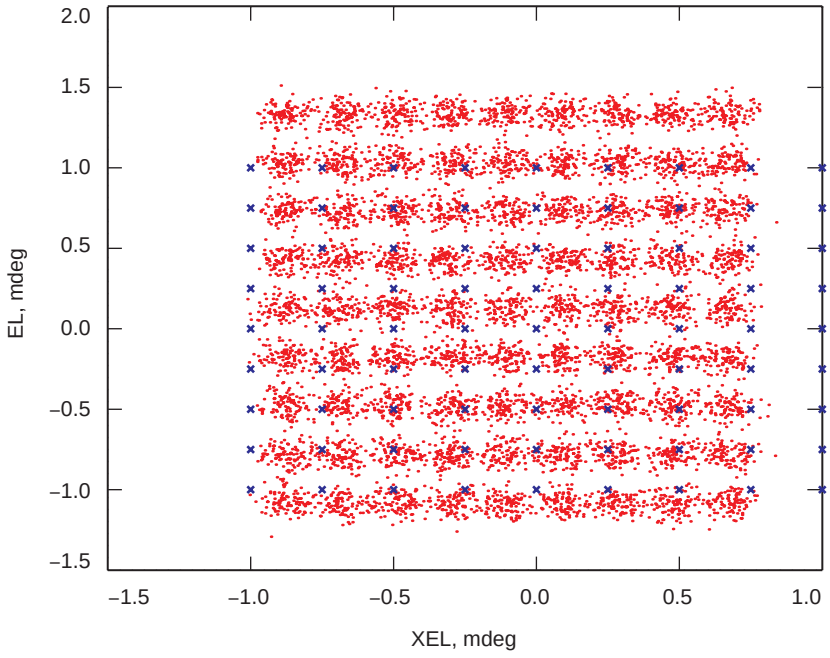

Fig. 26. Performance of the neural network retrained with the LMS algorithm at 45 deg.

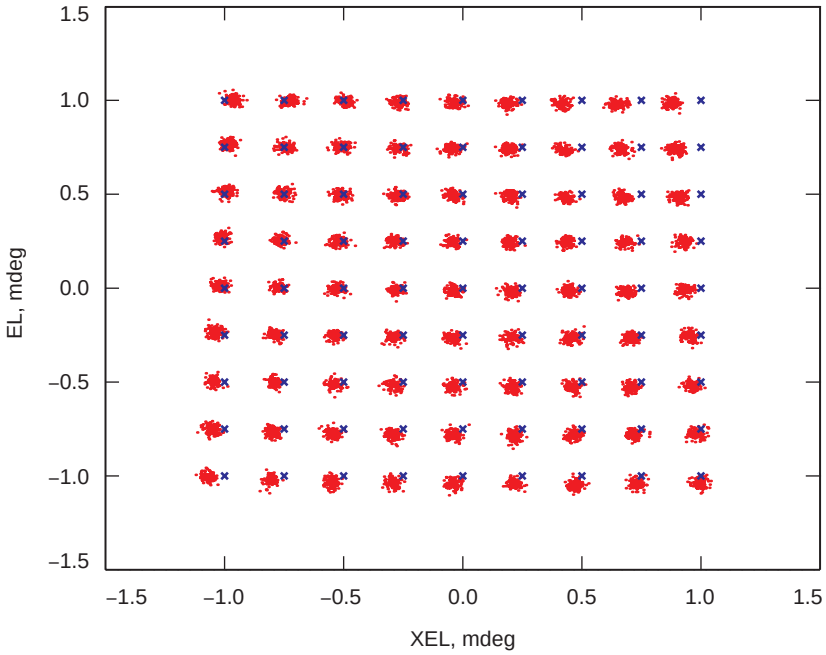

Fig. 27. Performance of the neural network retrained with the LMS algorithm at 75 deg.

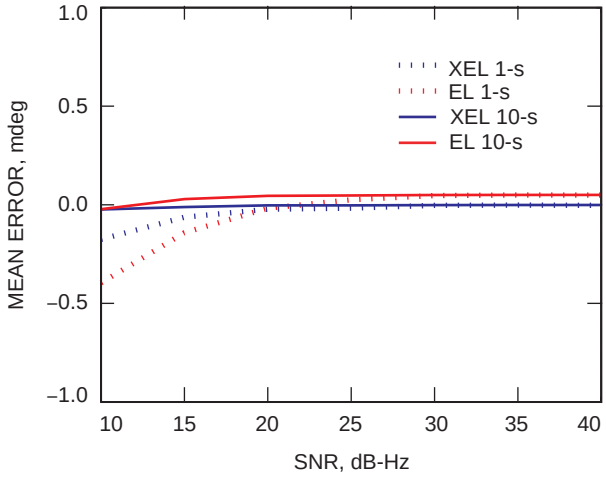

Fig. 28. Mean error after LMS retraining.

regenerate the neural networks, such regeneration is the most computationally expensive option and is not recommended unless significant changes in the antenna have occurred.

The only changes to the voltage-to-offset mapping simulated were linear distortions. In the real world, it is likely that nonlinear changes will appear in the voltage-to-offset mappings. Adaptive testing on more realistic changes in the antenna remains necessary.

G. Short-Term Adaptive Strategies

An SNR measurement algorithm that could accept antenna offsets given by the antenna encoders (devices that yield accurate measurements of the antenna's pointing offset relative to a fixed point)

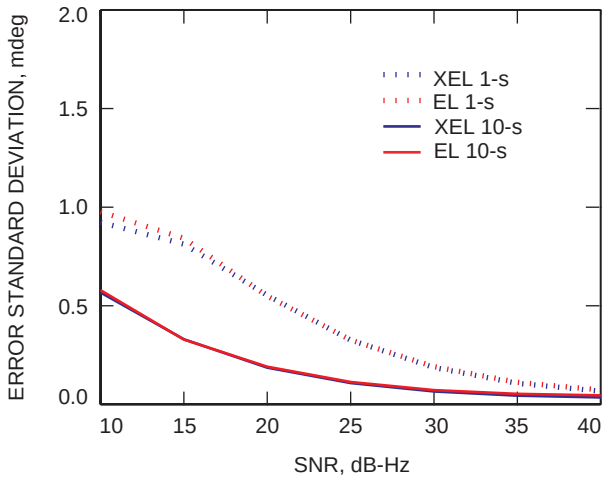

Fig. 29. Error standard deviation after LMS retraining.

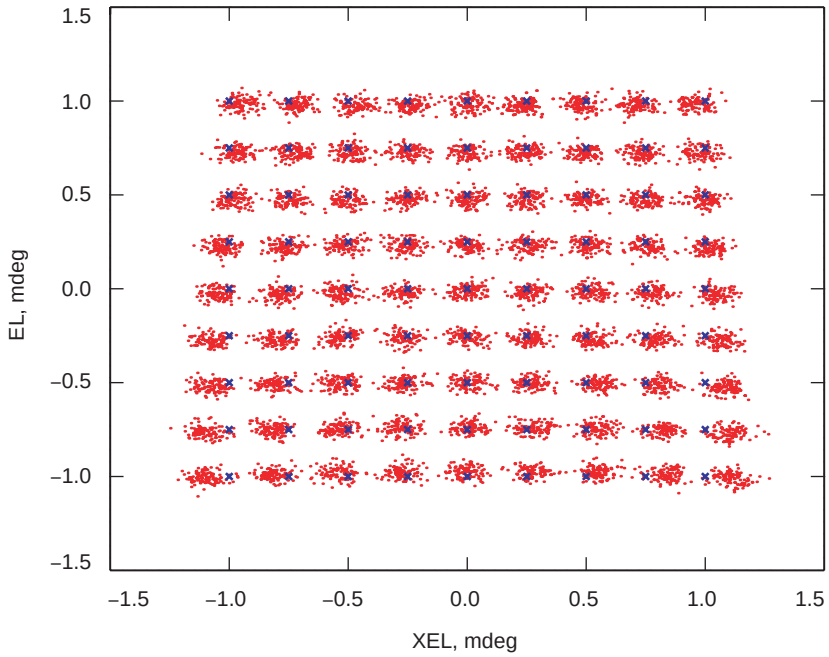

Fig. 30. Performance of the neural network retrained using the pseudo-inverse method at 15 deg.

together with the received SNR measurements could be used to determine the location of peak SNR in antenna offset space. By constructing an SNR surface, which shows SNR as a function of the antenna offset, we can gather information that would allow fine tuning of antenna pointing in real time, possibly improving upon the performance of existing algorithms. Furthermore, such information could be used to detect systematic pointing errors in both the least-squares and the RBF network systems, allowing them to be updated appropriately. For example, the bias weight in the RBF network's linear combiner can be updated using the mean pointing error determined by the SNR measurement algorithm, and similar updates could be performed on adaptive lookup tables for interpolated least-squares algorithms as well. The SNR measurement algorithm, therefore, would serve two purposes: it would provide real-time fine-pointing corrections and it would play a role in the long-term adaptation of the RBF network and of the adaptive lookup tables used by interpolated least-squares systems.

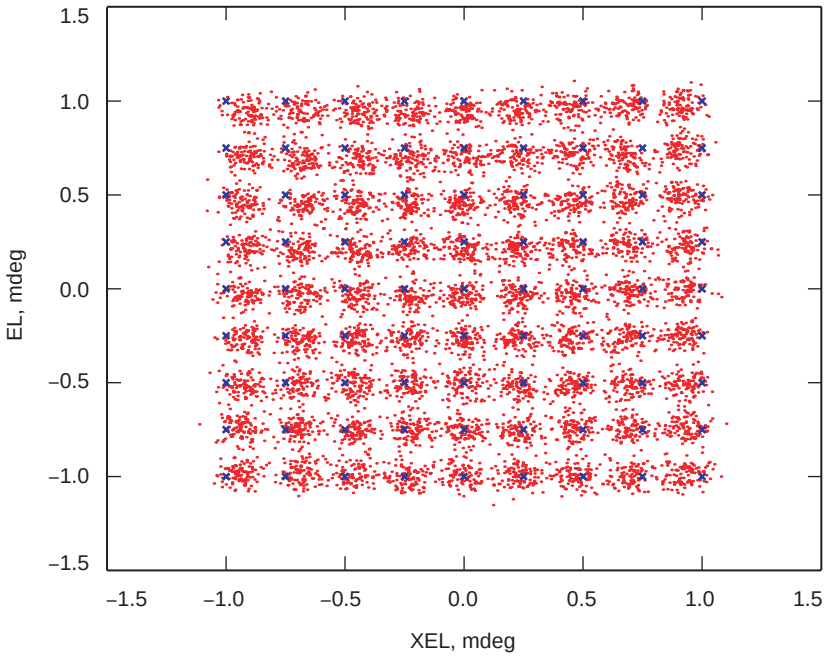

Fig. 31. Performance of the neural network retrained using the pseudo-inverse method at 45 deg.

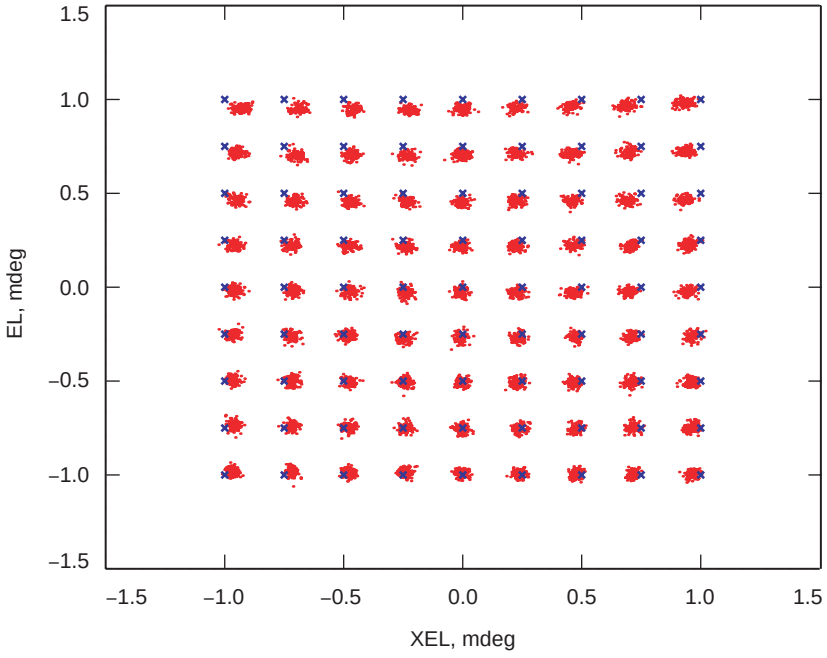

Fig. 32. Performance of the neural network retrained using the pseudo-inverse method at 75 deg.

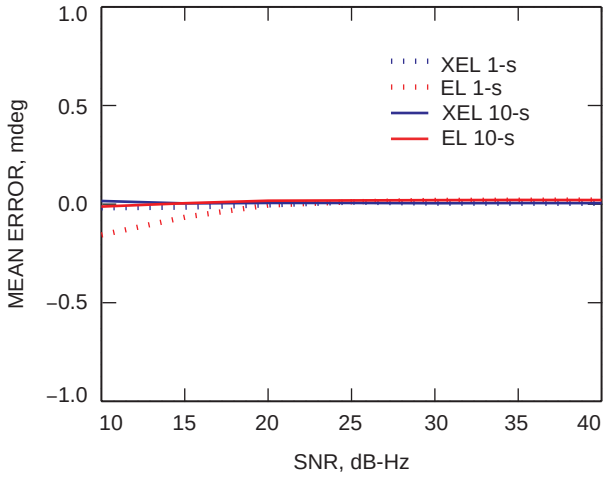

Fig. 33. Mean errors after pseudo-inverse retraining.

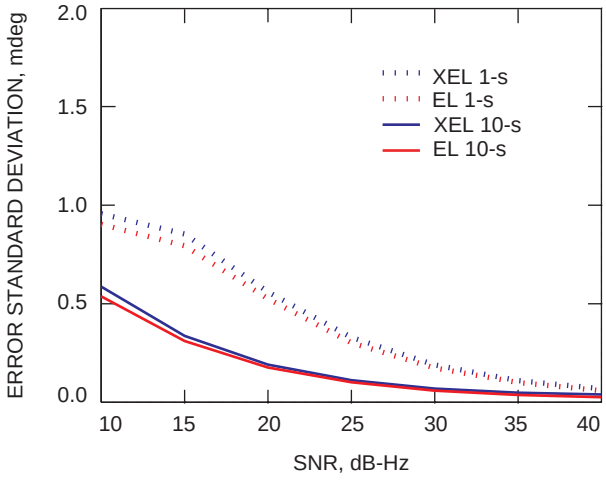

Fig. 34. Error standard deviations after pseudo-inverse retraining.

VI. Conclusions

Both interpolated least-squares and adaptive RBF network algorithms demonstrate clear potential for accurate pointing of the 70-m DSN antennas. Results indicate that the RBF network performs nearly as well as a quadratic interpolated least-squares algorithm for low-to-medium antenna elevation. At high antenna elevations, the quadratic interpolation method deteriorates for any significant offset from the on-source direction, making RBF networks especially attractive during high-elevation operations.

Adaptive tracking remains a topic of open research. It has been shown that adaptation of the linear combiner weights in the RBF network yields very good performance for simple stretch, shrink, and translational distortions in antenna offset space. More work remains to be done, and future topics include an adaptive interpolated least-squares algorithm and SNR-measurement-based fine tuning of the pointing estimate. Adaptive interpolated least-squares algorithms would use a data-windowing technique to average the latest data in order to continuously update the lookup table used by the least-squares system, allowing interpolated least-squares algorithms to operate adaptively as the antenna changes

with age. Since the interpolated least-squares algorithm is nearly optimal for low-to-medium elevations, adaptive least-squares algorithms will play an important role in future research, and adaptive tables would also serve as training sets for RBF network adaptation.

References

[1] R. Mukai, V. Vilnrotter, P. Arabshahi, and V. Jamnejad, "Computationally Intelligent Array Feed Tracking Algorithms for Large DSN Antennas," *The Telecommunications and Mission Operations Progress Report 42-141, January–March 2000*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–16, May 15, 2000.
http://tmo.jpl.nasa.gov/tmo/progress_report/42-141/141I.pdf

[2] V. Vilnrotter and D. Fort, "Demonstration and Evaluation of the Ka-Band Array Feed Compensation System on the 70-Meter Antenna at DSS 14," *The Telecommunications and Mission Operations Progress Report 42-139, July–September 1999*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–17, November 15, 1999.
http://tmo.jpl.nasa.gov/tmo/progress_report/42-139/139J.pdf

[3] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, March 1999.

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

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