UNLIMITED

RSRE
MEMORANDUM No. 4567

ROYAL SIGNALS & RADAR ESTABLISHMENT

A COMPARISON OF FEED-FORWARD NETWORKS AND MAXIMUM LIKELIHOOD ON A POINT-SOURCE LOCATION PROBLEM

Author: A R Webb

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

RSRE MEMORANDUM No. 4567

UNLIMITED

CONDITIONS OF RELEASE

0120143 308887

********** DRIC U

********** DRIC Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

Royal Signals and Radar Establishment
Memorandum 4567

A Comparison of Feed-forward Networks and Maximum Likelihood on a Point-source Location Problem

Andrew R. Webb

17th April 1991.

Abstract

The problem of point source location using a multi-beam focal-plane staring array radar is addressed. It is viewed as one in functional approximation in which the position of the source is regarded as a nonlinear function of the sampled radar image and it is required to construct an approximant, using a training set, which minimises the mean square error in the position estimate. The problem is also one of generalisation, since the expected operating conditions are likely to be corrupted by noise and this must be taken into account when designing the approximant. Two feed-forward network architectures are considered - a particular radial basis function network which arises as a consequence of the minimum mean square error solution and is appropriate when the signal-to-noise ratio is 'small' and a multilayer perceptron, chosen for high signal-to-noise ratio approximation. The errors in the position estimates for each of these approaches are compared with a maximum likelihood position estimation method. The maximum likelihood method gives better overall performance and has the advantage that it is not dependent on the signal-to-noise ratio.

INTENTIONALLY BLANK

Contents

INTENTIONALLY BLANK

1 Introduction

The problem addressed in this paper is one of multi-sensor data analysis. Data generated by a given sensor system represents a particular view of the scene under consideration. The signal processing problem is to provide a description of that scene, given some *a priori* knowledge of the scene characteristics and knowledge of the properties of the sensor. For example, in an active radar situation, the scene may be comprised of point scatterers, distributed scatterers, clutter, chaff and interference. One common form of *a priori* knowledge imposed is the special case that the scene can be represented spatially by a collection of point sources. There are many techniques for estimating the parameters of sources in a scene, all of which require some form of training of the system. This may be achieved by moving a single source around in the far field and recording the output of the system. If the outputs of all sensors in the system are sampled simultaneously, then we obtain a vector of numbers, the 'image vector', which gives a snapshot from the system for a given source position. All these vectors are collected together as columns of a matrix which forms a 'reference library' of signals expected from each incident direction. This library lies on a two–dimensional manifold, termed the *array manifold* [18, 23], within the space of sensor outputs.

The particular sensor system we consider is a focal–plane array radar. Focal–plane array technology provides a wide multiple–beam field of view with no moving parts and benefits from a high level of front–end circuit integration [1]. It uses a lens to provide multiple–beam coverage over a wide field of view, and a planar array of receivers, with no requirement for any beam–forming circuitry. Furthermore, the individual receivers that make up the array are very small and they can be designed so as not to contain any microwave circuitry. All of these factors combine to make the receiver and array architecture so simple that it is potentially possible to implement complete arrays within a small area of low–cost monolithic silicon. The two principal components of a focal–plane receiver front–end are a dielectric lens and an array of receivers. The lens system focuses incoming radiation on to the antenna array. The combined operation of the lens and receiver array provides a multiplicity of beams, each with its own direction of look. The radiation pattern of each beam depends on the lens aperture, the properties of the lens, the responses of the receivers and their poositions on the focal plane.

One of the main purposes of this paper is to show a possible use of adaptive feed–forward networks (or 'neural' networks) to the problem of point–source location using radar focal–plane arrays. Neural networks for sensor signal processing tasks are currently an area of considerable research [4, 25]. One particular area of interest is that of automatic target recognition [21] and the problems which have been addressed to date apply neural network techniques to data from a variety of sensor outputs including radar [2, 4, 6], sonar [8], infra–red and laser returns and these techniques have been used to identify various target types such as ships, aircraft, munitions, ground vehicles in a clutter environment, and terrain types. Other signal processing problems being addressed include bearing estimation [9, 13], multitarget tracking [15] and radar signal categorisation [20].

The advantage of an adaptive network solution to the problem of point–source location is that the network implicitly allows a parametrisation of the point–spread function or array manifold (by the weights in the network) which obviates the need to store the point–spread function explicitly. Also, with technology currently being developed there is the potential for an integrated solution on the focal plane of the system. There are other techniques which

may be used, however. A maximum likelihood approach to position estimation has been considered in [27] and indeed a neural network implementation of a maximum likelihood algorithm is described in [11]. The approach in this paper differs from that in [11] in that we are considering feed-forward architectures and we consider the effects of noise on the estimates in position.

The specific problem we shall consider in this report is the estimation of the position of a *single* source in the scene given its sampled image vector. For illustration purposes, we shall restrict the analysis and the numerical examples in this paper to linear arrays, though it applies equally to two-dimensional arrays. Section 2 describes the generation of an image vector and how a library of such vectors may be used in the problem of point source location. In Section 3 we consider the problem of deriving an approximation to a known functional transform which generalises to points not in the data set and which approximates the function in a minimum mean square error sense. Section 4 gives a brief description of adaptive feed-forward networks and methods of training such networks. Section 5 considers the application of the network to point source location, with the specific example of an idealised linear array of receivers in the focal-plane of an imaging system. The problem is one of generalisation. In a practical situation, the array outputs (the inputs to the feed-forward network) are likely to be corrupted by noise. Therefore, we wish to design a network, based on the training data characterising the array manifold (perhaps generated from a model of the imaging process or obtained during some calibration of the system), which generalises from the noiseless training data set to input vectors corrupted by noise. Two types of network are considered. One is a particular radial basis function network (see Section 4) appropriate when the expected noise "in operation" is large (a low signal-to-noise ratio). The second is a multilayer perceptron architecture designed for high signal-to-noise ratios. The performance of these networks is compared to a maximum likelihood approach. Finally, the paper concludes with a discussion of the results and a summary of the advantages and disadvantages of the use of a network for point source location and gives some suggestions for further work.

2 The Imaging Problem

The problem of point-source location may be posed as one of image restoration in which we desire to reconstruct a scene from a set of measurements of the image of the scene, given some knowledge of the imaging operation. This knowledge is often expressed in terms of the *point-spread function*, usually specified as a library of vectors. This library of vectors is generated from the outputs of an array of N sensors in the focal plane of an imaging system as follows.

The one-dimensional imaging equation relating a time-varying image $g(x;t)$ to the scene, $f(\xi;t)$ is given by a convolution equation of the form

$$g(x;t) = \int h(x;\xi) f(\xi;t) d\xi + n(x;t) \tag{1}$$

where $h(x;\xi)$ is the point-spread function of the imaging system and $n(x;t)$ is the noise in the degraded image.

When the image is sampled, the image is known at only a finite number of points in the image plane $(x_1, x_2, \ldots, x_N)$ corresponding, in the focal plane imaging problem, to the

array receiver positions and Equation (1) becomes

$$g_i(t) = \int h_i(\xi) f(\xi; t) d\xi + n_i(t) \tag{2}$$

where $g_i(t)$ is the value of the sampled image at position x_i at time t and N is the number of image sample positions (number of receivers). The function $h_i(\xi)$ is the response at the position x_i to a point source in the far-field as a function of position of that source. For an ideal, diffraction-limited, space-invariant imaging system (one which acts uniformly across image and object planes) (with a narrow slit as the aperture) the response $h_i(\xi)$ is given by

$$h_i(\xi) = \frac{sin[\Omega(x_i - \xi)]}{\pi(x_i - \xi)} \tag{3}$$

In the examples of Section 5, we take $\Omega = \pi$, so that sampling at the Nyquist rate, (π/Ω) gives unit spacing of the sample points.

If noise effects are absent then a point source of unit amplitude in the far field at a position (ξ_0) gives rise to an image vector

$$\boldsymbol{h}(\xi_0) \equiv (h_1(\xi_0), h_2(\xi_0), ..., h_N(\xi_0))^* \tag{4}$$

where * denotes vector transpose. The library of vectors used to characterise the imaging operation consists of a set of P such images of sources $(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_P)$ at P different positions in the scene (these images lie on a one–dimensional manifold, termed the *array manifold*, in the N–dimensional space of sensor outputs) together with the set of corresponding positions $\{\xi_i, i = 1, \ldots, P\}$. Thus the data points used to characterise the imaging operation are points $\{(\boldsymbol{h}_i, \xi_i), i = 1, \ldots, P\}$ in the space $\mathbb{R}^N \otimes \mathbb{R}$. In the terminology of feed–forward networks, this is referred to as the training set. The image vectors, $\boldsymbol{h}_i$, will be complex–valued in general, though in our examples we shall consider the ideal responses given by Equation (3) which gives rise to real–valued quantities.

The problem of image restoration is to reconstruct an object from its band–limited image, given knowledge of the point–spread function and some *a priori* knowledge concerning the object. The particular constraint that we consider here is that the scene consists of a single point source (of unknown amplitude, A) and the problem addressed is the determination of the *position* of the source given its image vector, $\boldsymbol{I}$, which is of the form

$$\boldsymbol{I} = A\boldsymbol{h} + \boldsymbol{n},$$

where $\boldsymbol{n}$ is a noise vector. Thus, the image vector is a scaled (by amplitude A, which may be a complex quantity) and corrupted (by additive noise) version of a vector $\boldsymbol{h}$, which lies on the array manifold, but which is not necessarily a member of the training set.

3 Minimum Mean–square Estimate

Before we discuss feed–forward networks in Section 4 and consider in particular their application to position estimation in Section 5, we shall present some functional approximation preliminaries. We view the problem of point–source location, given an image vector, as one

in nonlinear function approximation and generalisation. That is, we regard the position of the source in the scene as a nonlinear function of the image, with the form of the nonlinear function being specified by the training data. Also, we wish to generalise to data points not in the training set. This was the problem addressed in [26] and in this section we shall present some results relating to the approximation of a function $f(\boldsymbol{x})$, where $\boldsymbol{x}$ is a noiseless data sample, by a function $g(\boldsymbol{z})$, where $\boldsymbol{z}$ is a data sample corrupted by additive noise.

Suppose that we wish to approximate a transformation $\boldsymbol{f}$ from $\mathbb{R}^n$ to $\mathbb{R}^{n'}$. Let the approximation be given by $\boldsymbol{g}$ which is chosen so that the quantity V, defined by

$$V = \int\int |\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x} + \boldsymbol{\xi})|^2 p_n(\boldsymbol{\xi}) p(\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{\xi} \tag{5}$$

is a minimum, where $p_n(\boldsymbol{\xi})$ is the probability density function of a noise distribution in the space $\mathbb{R}^n$ and $p(\boldsymbol{x})$ defines the distribution of data points $\boldsymbol{x}$ in the space $\mathbb{R}^n$. Equation (5) defines the expected square error in the approximation when the data points in the domain of $\boldsymbol{f}$ are corrupted by additive noise, and may be written (for $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{\xi}$) as

$$V = \int\int (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{z}))^2 p_n(\boldsymbol{z} - \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{z}. \tag{6}$$

Minimising with respect to the function $\boldsymbol{g}$ gives the solution for $\boldsymbol{g}$ as

$$\boldsymbol{g}(\boldsymbol{z}) = \frac{\int \boldsymbol{f}(\boldsymbol{x}) p_n(\boldsymbol{z} - \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}}{\int p_n(\boldsymbol{z} - \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}} \tag{7}$$

This is the approximation to the function $\boldsymbol{f}$ for which the expected square error in the functional value, integrated over the domain of $\boldsymbol{f}$, is a minimum and generalises $\boldsymbol{f}$ to points $\boldsymbol{z}$ outside the distribution of the data points $\boldsymbol{x}$.

More generally, the minimum mean square estimate of $\boldsymbol{f}$ given $\boldsymbol{z}$ is the expected vector of the *a posteriori* density [7]

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{z}) = E[\boldsymbol{f}(\boldsymbol{x})|\boldsymbol{z}] &= \int \boldsymbol{f}(\boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{x} \\ &= \frac{\int \boldsymbol{f}(\boldsymbol{x}) p(\boldsymbol{z}|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}}{\int p(\boldsymbol{z}|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}} \end{aligned} \tag{8}$$

Note that the function $\boldsymbol{g}(\boldsymbol{z})$ may be defined over the whole space $\mathbb{R}^n$, whereas the data points $\boldsymbol{x}$ may lie on a reduced dimension manifold, X, in $\mathbb{R}^n$ (as specified by the probability density function, $p(\boldsymbol{x})$). Thus, the approximation to $\boldsymbol{f}$, $\boldsymbol{g}(\boldsymbol{z})$, is defined for values of $\boldsymbol{z}$ which do not necessarily lie on the manifold, X. This is important in many applications in which noise will corrupt data points, $\boldsymbol{x}$, to give values $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{\xi}$ which lie outside the domain of $\boldsymbol{f}$. In these situations it is not sufficient to interpolate the training set $\{(\boldsymbol{x}_i, \boldsymbol{f}_i), i = 1, \ldots P\}$ without due regard to defining the mapping for points outside the manifold.

The minimum mean-square approximation derived above provides a biased estimate, in that for a data point, $\boldsymbol{x}_0$, the mean of the estimate (the average over all perturbations $\boldsymbol{\xi}$ to $\boldsymbol{x}_0$) is not necessarily equal to the functional value $\boldsymbol{f}(\boldsymbol{x}_0)$, *i.e.*

$$\int \boldsymbol{g}(\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{x}_0) d\boldsymbol{z} \neq \boldsymbol{f}(\boldsymbol{x}_0) \tag{9}$$

where $\boldsymbol{z} = \boldsymbol{x}_0 + \boldsymbol{\xi}$. In some practical situations it may be advantageous to have an *unbiased* estimate so that integration may be performed after the functional transformation, *i.e.* we need to produce an approximation $\boldsymbol{g}(\boldsymbol{z})$ which is defined for all noise perturbations and which, for inputs $\boldsymbol{z} = \boldsymbol{x}_0 + \boldsymbol{\xi}$, if averaged will tend to $\boldsymbol{f}(\boldsymbol{x}_0)$, the true value in the absence of noise. An approach for finding such an unbiased approximation using Lagrange multipliers is given in [26].

3.1 Radial Basis Function Approximation

For a function $\boldsymbol{f}$ defined by a finite set of points $\{(\boldsymbol{x}_i, \boldsymbol{f}_i),\ i = 1, \ldots, P\}$ in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$, then provided that the integrands in Equation (7) are sufficiently smooth, the solution $\boldsymbol{g}$ may be approximated by $\hat{\boldsymbol{g}}$ given by

$$\hat{\boldsymbol{g}}(\boldsymbol{z}) = \frac{\sum_{i=1}^{P} \boldsymbol{f}_i\, p_n(\boldsymbol{z} - \boldsymbol{x}_i)}{\sum_{i=1}^{P} p_n(\boldsymbol{z} - \boldsymbol{x}_i)} \tag{10}$$

or

$$\hat{\boldsymbol{g}}(\boldsymbol{z}) = \sum_{i=1}^{P} \boldsymbol{f}_i\, \tilde{p}_n(\boldsymbol{z} - \boldsymbol{x}_i) \tag{11}$$

where $\tilde{p}_n(\boldsymbol{z} - \boldsymbol{x}_i)$ is defined by

$$\tilde{p}_n(\boldsymbol{z} - \boldsymbol{x}_i) = \frac{p_n(\boldsymbol{z} - \boldsymbol{x}_i)}{\sum_{i=1}^{P} p_n(\boldsymbol{z} - \boldsymbol{x}_i)} \tag{12}$$

Equation (11) is identical in form to radial basis function approximations [3] in that the approximating functional is a linear combination of (specified) nonlinear functions of the difference between a data point, $\boldsymbol{z}$ and a 'centre'. In this case the nonlinear basis functions are determined by the noise probability density function, the centres by the data points $\boldsymbol{x}_i$, and the weights are the function values, $\boldsymbol{f}_i$ at the centres. Thus a radial basis function network structure arises as a natural consequence of the minimum variance solution. For example, for a Gaussian noise model with diagonal covariance matrix with equal diagonal elements σ^2,

$$\hat{\boldsymbol{g}}(\boldsymbol{z}) = \frac{\sum_{i=1}^{P} \boldsymbol{f}_i \exp[-\frac{1}{2\sigma^2}|\boldsymbol{z} - \boldsymbol{x}_i|^2]}{\sum_{i=1}^{P} \exp[-\frac{1}{2\sigma^2}|\boldsymbol{z} - \boldsymbol{x}_i|^2]}. \tag{13}$$

Note that in order to derive the function $\boldsymbol{g}$ which approximates $\boldsymbol{f}$ and generalises to unseen data, we have not assumed a specific functional form, nor a smoothness condition (as in a regularisation theory approach). We have assumed that we know how to perform the mapping if there were no noise (noiseless training data) and assumed a minimum mean square error measure. A consequence of this is the radial basis function nature of the solution. However, we do need to know the noise distribution. If we were to assume that it is Gaussian with diagonal covariance matrix with equal elements, then we would need to specify the noise variance, σ^2, on the test data.

The function $\hat{\boldsymbol{g}}$ will provide a good approximation to the exact minimum mean-square solution, $\boldsymbol{g}$, if the standard deviation of the noise is large compared to the distance between sample points, $\boldsymbol{x}_i$.

3.2 Perturbation Analysis for High Signal-to-Noise Ratios

The solution for the minimum variance approximation to a known function, $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ is given by Equation (7). When the functional transformation is specified only by points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$, then this minimum variance solution may be approximated by a summation which takes the form of a radial basis function network with nonlinear functions being (normalised) noise probability density functions. This summation will be a good approximation to the minimum variance solution provided that the standard deviation of the noise distribution is large compared to the spacing between samples, $\boldsymbol{x}_i$. In a low noise situation (where the standard deviation of the noise distribution is small compared to the distance between sample points), the approximation $\hat{\boldsymbol{g}}(\boldsymbol{z})$ to $\boldsymbol{g}(\boldsymbol{z})$ will be accurate only in the region of the sample points and at intermediate values will give a very poor approximation. Therefore, we need to specify a model for the approximation to $\boldsymbol{f}(\boldsymbol{x})$, or a constraint in the form of a regularisation term, in order to describe how the function varies between sample points.

Let us assume that we have a parameterised model for the approximation to $\boldsymbol{f}$. In the following section, we shall consider a specific model (namely a feed-forward network), but at the moment there is no restriction to its form other than it is a continuous function, $\boldsymbol{g}$, of the data $\boldsymbol{z}$ with continuous first derivatives. First of all we shall calculate the perturbation to the error between the actual values, $\boldsymbol{f}_i$ and the approximate values due to noise on the data points.

Let $\{(\boldsymbol{x}_i, \boldsymbol{f}_i,\ i = 1, \ldots, P\}$ denote the set of points describing the mapping $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$. For a given data value, $\boldsymbol{x}_p$, let $E_p \equiv E(\boldsymbol{x}_p)$ be the error between the approximation to $\boldsymbol{f}(\boldsymbol{x})$ and the desired value, $\boldsymbol{f}_p$ for the pth pattern, $\boldsymbol{x}_p$. Often, the total error, is given by

$$E_T = \frac{1}{P}\sum_{p=1}^{P} E_p = \frac{1}{P}\sum_{p=1}^{P} E(\boldsymbol{x}_p). \tag{14}$$

with $E(\boldsymbol{x}_p)$ being the square of the error for pattern, $\boldsymbol{x}_p$, between the desired value (termed the 'target' values in a feed-forward network framework), and the approximation, giving E_T as the sum-square error between the approximations and the desired values. However, in the analysis which follows we impose no such restriction.

If the input patterns are corrupted by noise, *i.e.* they are of the form $\boldsymbol{x}_p + \boldsymbol{n}$, where the noise vector $\boldsymbol{n}$ has the property that $\langle \boldsymbol{n}\boldsymbol{n}^* \rangle = \sigma^2 \boldsymbol{I}$, ($\boldsymbol{I}$ is the $n \times n$ identity matrix) then it is shown in Appendix B that the expected error at the output, $\langle E_T \rangle$ may be written

$$\langle E_T \rangle = \frac{1}{P}\sum_{p=1}^{P} E(\boldsymbol{x}_p) + \frac{\sigma^2}{2P}\sum_{p=1}^{P} Tr(\boldsymbol{H}^p). \tag{15}$$

The first term in the expression is the error in the approximation when there is no noise on the data. The second term is a second derivative quantity proportional to the noise variance σ^2. For $\sigma^2 = 0$, $\langle E_T \rangle$ reduces to the usual error term in the absence of noise. Thus, if we have a mapping $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ defined by points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$ in which the data points in $\mathbb{R}^n$ are corrupted by additive noise with zero mean and variance σ^2 (sufficiently small so that the higher order terms in the Taylor expansion may be neglected), then minimising the error over all patterns and over the noise distribution with respect to the parameters of the approximating function, $\boldsymbol{g}$, is equivalent to minimising a modified error term defined on the

patterns in the absence of noise. Equation (15) shows that the effects of noise on the test data can be compensated for by training an approximant with a modified error criterion. A different approximation to $\boldsymbol{f}$ can be derived for different values of the noise variance, σ^2.

The two terms in Equation (15) may be regarded as the usual error metric plus a regularisation or stabilising term with regularisation parameter σ^2, the variance of the noise on the inputs. For the sum-squared error criterion, the second term in Equation (15) may be written as

$$\frac{\sigma^2}{P}\sum_{p=1}^{P}\left(\|\boldsymbol{J}^p\|^2 - (\boldsymbol{f}_p - \boldsymbol{g}(\boldsymbol{x}_p))^*\boldsymbol{q}^p\right) \tag{16}$$

where the $n' \times n$ matrix $\boldsymbol{J}^p$ is the Jacobian

$$\boldsymbol{J}^p_{ij} = \left.\frac{\partial g_i}{\partial x_j}\right|_{\boldsymbol{x}_p} \tag{17}$$

representing the derivative of the ith component of the approximation with respect to the jth input, evaluated for pattern $\boldsymbol{x}_p$. The vector $\boldsymbol{q}^p = (q^p_1, q^p_2, \ldots, q^p_n)^*$ is a vector of second derivative terms, with kth component

$$q^p_k = \sum_{i=1}^{n}\left.\frac{\partial^2 g_k}{\partial x_i^2}\right|_{\boldsymbol{x}_p} \tag{18}$$

evaluated for the pth pattern.

3.3 Summary

It is appropriate at this stage to summarise the results of this section.

1. Suppose that we have a known function, $f(\boldsymbol{x})$ which we wish to approximate. In the problem considered in this paper, $\{\boldsymbol{x}\}$ is the set of images of a point source in the absence of noise and $f(\boldsymbol{x})$ is the position of the source.
2. Suppose that we wish to approximate in a least squares sense the function $f(\boldsymbol{x})$ by a function $g(\boldsymbol{z})$ which is defined for points $\boldsymbol{z}$ outside the set $\{\boldsymbol{x}\}$. For example, $\boldsymbol{z}$ may be the image of a source corrupted by additive noise, *i.e.* $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{n}$

then

- the solution for $g(\boldsymbol{z})$ is given by

$$\boldsymbol{g}(\boldsymbol{z}) = \frac{\int \boldsymbol{f}(\boldsymbol{x})p_n(\boldsymbol{z} - \boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}}{\int p_n(\boldsymbol{z} - \boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}} \tag{19}$$

- This may be approximated by a finite sum

$$\hat{\boldsymbol{g}}(\boldsymbol{z}) = \frac{\sum_{i=1}^{P} \boldsymbol{f}_i\, p_n(\boldsymbol{z} - \boldsymbol{x}_i)}{\sum_{i=1}^{P} p_n(\boldsymbol{z} - \boldsymbol{x}_i)} \tag{20}$$

provided that the sample spacing of the points $\boldsymbol{x}_i$ is small compared to the standard deviation of the noise probability density function.

- If this is not so (the noise is small), then we assume a particular parametric form for $g(\boldsymbol{z})$ and choose the parameters which minimise an augmented sum-square error measure. This is equivalent to training on data representative of the operating conditions. That is, we may simulate the effects of noise by using the noiseless data with a modified error criterion.

4 Feed-forward Adaptive Networks.

Connectionist models based on feed-forward networks (for example, multilayer perceptrons (MLPs) [22] and radial basis function networks [3] (RBFs)) have been used with some success when operating as static pattern classifiers on a wide range of problems. Such networks perform a nonlinear transformation from an n-dimensional input space to the n'-dimensional output space via a characterisation space defined by the outputs of the (final layer of) hidden units in which a specific feature extraction criterion is maximised [17, 29]. This feature extraction criterion may be viewed as a nonlinear multidimensional generalisation of Fisher's linear discriminant function. Training the network for a pattern classification task consists of presenting data vectors as input, together with class labels at the output of the network, suitable coded, and minimising an error criterion. For a 1-from-n' target coding scheme, and the usual sum-square error criterion, the outputs of a trained network approximate the Bayes discriminant vector, giving the probability of a class given the input to the network [17].

An alternative viewpoint to the pattern classification description on the operation of adaptive, feed-forward layered networks such as the multilayer perceptron is that they perform well for certain tasks by exploiting their modelling flexibility to create an implicit interpolation surface in a high-dimensional space [3, 16]. In fact, it may be shown that multilayer feed-forward networks with a single hidden layer are universal approximators in that an arbitrary function can be approximated arbitrarily well [10, 24]. However, in a practical problem, the mapping we wish to approximate is not known continuously but it is usually defined by a finite set of points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$ defined by a training set. Specifically, in mapping a finite set of P, n dimensional 'training' patterns to the corresponding n' dimensional 'target' patterns, $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ one may think of this map as being generated by a 'graph' $\Gamma \subset \mathbb{R}^n \otimes \mathbb{R}^{n'}$. The input and target pattern pairs are points on this graph. The learning phase of adaptive network training corresponds to the optimisation of a fitting procedure for Γ based on knowledge of the data points. This is curve fitting in the generally high dimensional space $\mathbb{R}^n \otimes \mathbb{R}^{n'}$. Thus *generalisation* becomes synonymous with *interpolation* along the constrained surface which is the 'best' fit to Γ [31].

If there is noise in the expected operating conditions (on the test set) then this must be taken into account when designing a fitting surface to the training data points. This was the problem addressed in [26] in which it was shown how to construct a fitting surface which gives the expected value of the observation in $\mathbb{R}^{n'}$ given the data sample in $\mathbb{R}^n$.

The problem of point source location is one of generalistaion [26] in that we wish to generalise to data points (scaled and corrupted by noise) which are not in the training

set. In this section we give a brief description of the structures we shall use to process the outputs of a focal-plane array radar.

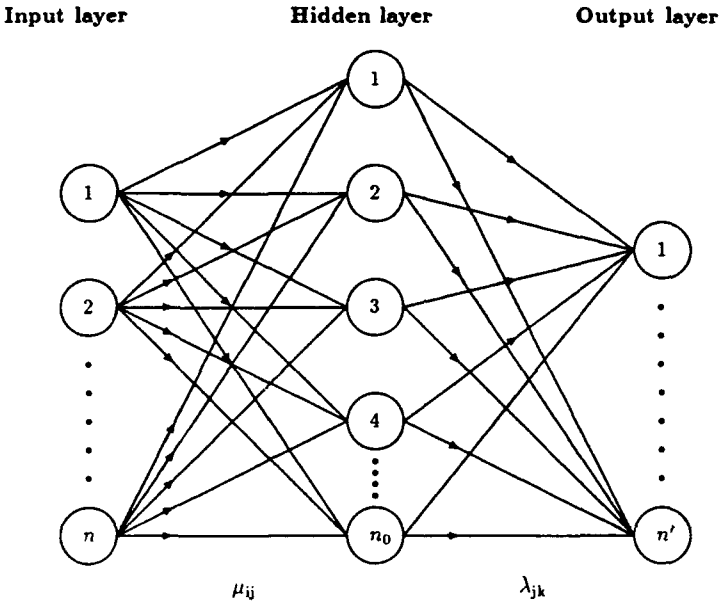

Figure 1: A schematic diagram of the standard feed forward adaptive layered network geometry considered in this paper.

The structure of a standard layered network model is depicted in Figure 1. It is envisaged that input data may be represented by an arbitrary (real-valued) n-dimensional vector, $\boldsymbol{x}$, or an ordered sequence of n real-valued numbers, $\{x_i;\ i = 1,\ldots,n\}$. Thus there are n independent input nodes to the network which accept each input data vector. Each input node is totally connected to a set of n_0 'hidden' nodes (hidden from direct interaction with the environment). Associated with each link between the i-th input node and the j-th hidden node is a scalar μ_{ij}. Usually, the fan-in to a hidden node takes the form of a hyperplane: the input to node j is of the form $\theta_j = \sum_{i=1}^{n} x_i\mu_{ij} = \boldsymbol{x}^*\boldsymbol{\mu}_j$ where $\boldsymbol{\mu}_j$ is the vector of n scalar values associated with hidden node j and * denotes transpose. The rôle of each hidden node is to accept the value provided by the fan-in and output a value obtained by passing it through a (generally, though not necessarily) nonlinear transfer function,

$$\phi_j = \phi(\mu_{0j} + \theta_j) = \phi_j(\mu_{0j} + \boldsymbol{x}^*\boldsymbol{\mu}_j) \tag{21}$$

where μ_{0j} is a local 'bias' associated with each hidden node. In principle, the input data vector may be an n-dimensional *complex*-valued vector with the nonlinearity defined to map complex input to real-valued output. However, in this paper, we shall consider only input data vectors which are real-valued.

The hidden layer is fully connected to a set of n' output nodes corresponding to the components of an n' dimensional output space. The strength of the connection from the j-th hidden node to the k-th output node is denoted λ_{jk} and thus the value received at the k-th output node is a weighted sum of the output values from all of the hidden nodes, $o_k = \sum_{j=1}^{n_0} \lambda_{jk}\phi_j$.

In general the output from the k-th output node will be a nonlinear function of its input, $o_k = \Phi_k(\lambda_{0k} + \boldsymbol{\lambda}_k^* \boldsymbol{\phi})$ where λ_{0k} is a 'bias' associated with that output node.

Thus the networks provide a transformation mapping from an n-dimensional input space to an n'-dimensional output space via an intermediate characterisation space. This mapping is totally defined by the topology of the network (in particular, how many hidden units are employed) once all the nonlinear transfer functions are specified and the set of weights and biases $\{\boldsymbol{\lambda}, \boldsymbol{\mu}\}$ have been determined. This set of weights and biases is found by a 'training' procedure.

Networks performing a transformation from an n-dimensional input space to an n'-dimensional output space using more than one intermediate hidden layer have been considered by some workers [19], but we shall restrict our attention in this paper to networks with a single hidden layer.

The network will operate once a set of weight values $\{\lambda_{jk}, \mu_{ij}\}$ has been determined. This set is conditional upon training data presented in the form of representative input and corresponding target output patterns. The set of parameters $\{\lambda_{jk}, \mu_{ij}\}$ is chosen so that the actual outputs of the network, $\{\boldsymbol{o}^p, p = 1, 2, \ldots P\}$, for a given set of inputs, $\{\boldsymbol{x}^p, p = 1, 2, \ldots P\}$, are 'close' in some sense to the desired target values, $\{\boldsymbol{t}^p, p = 1, 2, \ldots P\}$. Usually, this error criterion is a sum-of-squares error of the form

$$E = \sum_{p=1}^{P} \| \boldsymbol{t}^p - \boldsymbol{o}^p \|^2 \tag{22}$$

where the summation runs over all the patterns in the training set. Using the Euclidean distance function and expressing the outputs in terms of the set of weights and biases and the inputs, the error may be written explicitly as a function of the set $\{\lambda_{jk}, \mu_{ij}\}$. For instance, in the case of the standard multi-layer perceptron, this error may be expressed as

$$E = \sum_{p=1}^{P} \sum_{k=1}^{n'} \left\{ t_k^p - \Phi_k\left(\lambda_{0k} + \sum_{j=1}^{n_0} \phi_j[\mu_{0j} + \sum_{i=1}^{n} x_i^p \mu_{ij}]\lambda_{jk}\right) \right\}^2 \tag{23}$$

If the training data is not representative of the test data and we wish to derive an approximation to the mapping from $\mathbb{R}^n$ to $\mathbb{R}^{n'}$ defined by the training data for which the sum-squared error in operation is a minimum, then the error function used during training must be modified to take account of the discrepancies between training and operating conditions [26].

The expressions for the error, ((22) or the form (15), modified to take account of expected noise on the data) are differentiable nonlinear functions of the parameters and the aim of any training procedure is to find a minimum of this function. Therefore, some strategy for nonlinear function minimisation must be employed. Of course, a global minimum cannot

be guaranteed. Nevertheless, it may be possible to obtain a good local minimum. Also, not only do we require a good solution, but it must be obtained 'within a reasonable timescale'. Schemes which find a good solution a small percentage of the time, but are very fast, may be preferable to one which finds a good solution on most occasions, but takes a long time to do so. Optimisation strategies for nonlinear functions have been discussed in previous papers [30, 28]. These were applied to the training of adaptive feed-forward networks and various example problems considered. For the problem of point source location using a 4×4 focal-plane array, the best solution (in terms of the smallest mean error on test) was obtained using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimisation scheme. This is the method which we shall use in Section 5.3.

Testing the network consists of applying the trained network to patterns not previously used as part of the training set and comparing the outputs with the labels corresponding to those patterns. It is not sufficient to consider how closely the network models the training set alone since, if it models the training set too well, the network may not have captured the underlying structure of the data and be unable to generalise to unseen data.

5 Feed-forward Network Estimation of Source Position

In this section, we consider the application of feed-forward adaptive networks to point source location using focal-plane arrays. The method may be applied to any array of sensors where the image response function may be characterised by an array manifold. However, in order to be specific, we have confined our study to the focal-plane situation and one idealised array in particular, namely a 5×1 array of elements, each with a $sin(x)/x$ shape point-spread function (Equation 3). Thus, the array manifold consists of a set of real-valued vectors. In each example, the distance between adjacent elements in the focal-plane is unity, giving samples of the image at Nyquist rate. Figure 2 illustrates the response of each receiving element to a point source in the far field for the linear array. The distance between the peak of a response and the first null is termed the "beamwidth" and is equal to unity for these examples.

Section 5.1 describes the data used for training and testing the network. Sections 5.2 and 5.3 describe feed-forward network estimators of position. Section 5.4 assesses a maximum likelihood approach. This provides a reference by which to judge the feed-forward network technique. Section 5.5 gives results for the bias in the estimate of the position of a source as a function of position for both the linear and square arrays, and compares the results with the maximum likelihood method.

5.1 Generation of Data

Training and test sets have been generated for the linear array, with each set consisting of a set of images of single point sources of unit amplitude (used as input to a network), together with the source positions (taken to be the target data). Thus, the input dimension of the network, n is taken equal to the dimension of the receiver array, N. For all experiments, the size of the linear array was fixed to contain 5 receivers at Nyquist spacing in the focal-plane. Thus, the set of input data, $\{\boldsymbol{x}^p\}$ is a set of representative images, $\{\boldsymbol{h}(\boldsymbol{\theta}^p)\}$, with the corresponding targets, $\boldsymbol{t}^p$ being the positions $\boldsymbol{\theta}^p$.

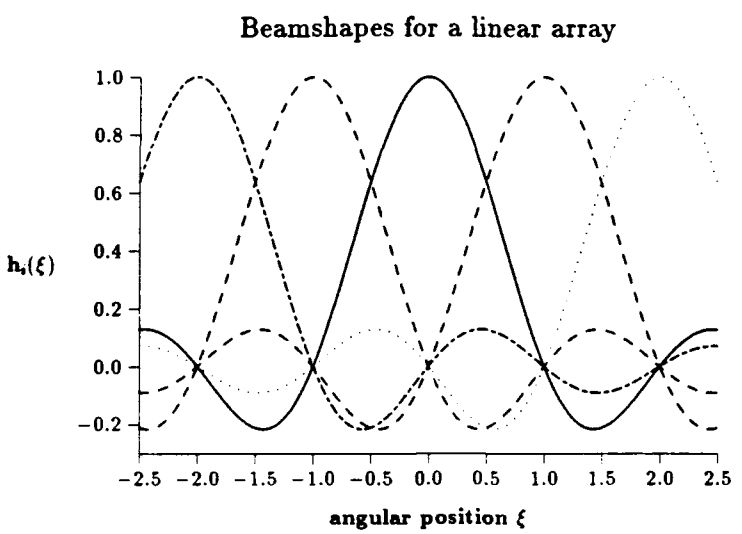

Figure 2: Response of each receiving element to a point source in the far–field for a 5×1 linear array of receivers in the focal–plane of an imaging system.

For the linear array considered, the images of a single source are calculated using Equation 3 at 101 different positions, equally spaced across the field of view of the array from -2.5 to 2.5 (at a spacing of $\frac{1}{20}$). For the test data, the images of a single source at 200 positions chosen randomly between -2.0 and 2.0 are taken as input with the source position as target.

The focal-plane array illustration described is highly idealised. In general, the array manifold, and the image vectors, would be complex vectors due to the phase of the source and the relative phase between receivers being a function of source position and therefore some method of incorporating complex vectors into a feed-forward network would have to be considered. This is not a difficult task, but for our purposes we shall restrict the example to considering real vector inputs only.

5.2 Radial Basis Function Approximation

We now derive, using the training data, an estimate of source position which is a nonlinear function of the measured image vector, $\boldsymbol{z}$. A naïve application of Equation (10) (with the $\boldsymbol{x}_i$ taken to be the data points and the f_i the target points) is inappropriate for the point source location problem. This is because, for a single point source in the scene, the measured image, $\boldsymbol{z}$, is not simply a point $\boldsymbol{x}$ on the array manifold corrupted by noise, but it is a *scaled* version of a point on the array manifold corrupted by noise, *i.e.*

$$\boldsymbol{z} = A\boldsymbol{x} + \boldsymbol{n} \tag{24}$$

where A is the amplitude of the source. Thus,

$$p(\boldsymbol{z}|\boldsymbol{x}, A) = p_n(\boldsymbol{z} - A\boldsymbol{x}), \tag{25}$$

where p_n is the noise probability density function.

The solution for $g(z)$ which minimises the variance now involves the prior probability density function of the amplitude, A,

$$g(z) = \frac{\int\int f(x)p_n(z - Ax)p(A)p(x)dxdA}{\int\int p_n(z - Ax)p(A)p(x)dxdA} \tag{26}$$

For a Gaussian noise process, with diagonal covariance matrix with equal diagonal elements, σ^2,

$$p_n(n) = \frac{1}{(2\pi)^{n/2}\sigma}\exp[-\frac{1}{2\sigma^2}|n|^2] \tag{27}$$

and assuming that $p(A)$ is uniformly distributed, then integrating (over $(0, \infty)$) with respect to A gives

$$g(z) = \frac{\int f(x)s(x, z)p(x)dx}{\int s(x, z)p(x)dx} \tag{28}$$

where

$$s(x, z) = \exp\left\{-\frac{1}{2\sigma^2}z^*(I - \hat{x}\hat{x}^*)z\right\}\frac{1}{|x|}\left(1 \pm \mathrm{erf}(\frac{1}{\sqrt{2\sigma^2}}|z^*\hat{x}|)\right) \tag{29}$$

where the + sign is taken if $z^*\hat{x} > 0$, and the minus sign if $z^*\hat{x} < 0$.

Approximating the integrals with respect to x by a summation over the training set (this implicitly assumes that all angles are equally likely since the training data is sampled uniformly in angle space)

$$\hat{g}(z) = \frac{\sum_{p=1}^{P} f(x_p)s(x_p, z)}{\sum_{p=1}^{P} s(x_p, z)} \tag{30}$$

This approximation is valid provided that the function $s(x, z)$ is sampled on a scale which is small compared to the standard deviation, σ; that is we require

$$|\hat{x}_p - \hat{x}_{p+1}| \lesssim \frac{\sigma^2}{|z|^2} \tag{31}$$

5.3 Multilayer Perceptron for High Signal–to–noise Ratio

In a high signal–to–noise ratio situation, the approximation given by Equation (30) becomes increasingly invalid. Therefore, we choose to approximate the function by a particular transformation and determine the parameters by some appropriate minimisation procedure. The particular functional form we have chosen is a feed–forward network having a single hidden layer with input in the form of a hyperplane and a nonlinear transfer function $\phi(z) = 1/(1 + e^{-z})$. In fact, it may be shown that multilayer feed–forward networks with a single hidden layer are universal approximators in that an arbitrary function can be approximated arbitrarily well [10, 24]. However, in a practical problem, the mapping we wish to approximate is not known continuously but it is usually defined by a finite set of points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$ defined by a training set. The output nodes are taken to be linear functions, $\Phi(z) = z$.

As in the radial basis function example above, we wish to define an approximant which not only generalises to noise data not in the training set, but also is relatively insensitive to source amplitude. Therefore we choose to normalise the data vectors to be of unit magnitude on input to the network. This removes, at least in a high signal-to-noise situation, the effect of fluctuations of image vector magnitude due to source amplitude fluctuations. Thus the network used is that depicted in Figure 3 : an input normalisation layer, a hidden layer and

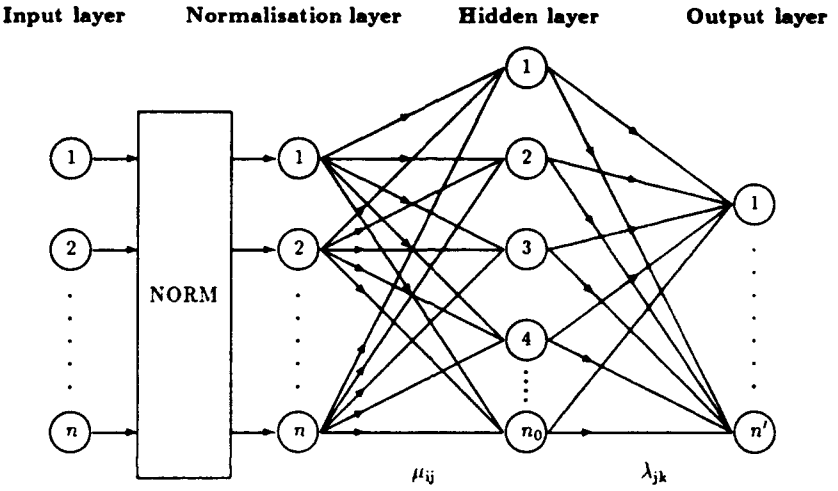

Figure 3: A feed forward adaptive layered network with an input normalisation layer.

a linear output layer.

For a multilayer perceptron with a single hidden layer and the sum-square error criterion, the regularisation term, Equation (16), may be written in terms of the weights using the results that

$$J^r_{ij} = \frac{\partial g}{\partial x_j}\Big|_{\boldsymbol{x}_p} = \sum_{k=1}^{n_0} \lambda_{ik} h^k_p (1 - h^k_p) \sum_{l=1}^{n} \mu_{kl} \frac{\partial x'_l}{\partial x_j}\Big|_{\boldsymbol{x}_p} \tag{32}$$

and

$$q^p_k = \sum_{i=1}^{n} \frac{\partial^2 g_k}{\partial x_i^2}\Big|_{\boldsymbol{x}_p}$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{n_0} \lambda_{kj} \left\{ h^j_p (1 - h^j_p)(1 - 2h^j_p) \left(\sum_{l=1}^{n} \mu_{jl} \frac{\partial x'_l}{\partial x_i}\Big|_{\boldsymbol{x}_p} \right)^2 + h^j_p (1 - h^j_p) \sum_{l=1}^{n} \mu_{jl} \frac{\partial^2 x'_l}{\partial x_i^2}\Big|_{\boldsymbol{x}_p} \right\} \tag{33}$$

where h^p_j is the output of the jth hidden node for input pattern $\boldsymbol{x}_p$ and x'_j represents the jth component of the normalisation layer. The scalar quantities λ_{ij} and μ_{jk} are the weights between the ith output node and the jth hidden node, and between the jth hidden node and the kth input node respectively.

For a given value of n_0 and a given set of training data, the network was trained to minimise the sum-squared error using the procedure described in Section 4. For the source location problem, the error between the outputs of the network and the targets which is minimised has a physical interpretation: it is the sum of the square of the error in position estimation. Initially, the values of the network weights were chosen randomly from a uniform distribution on $(-1.0, 1.0)$. Then the BFGS nonlinear optimisation strategy was used to find the solution for the weights for which the mean square error at the output of the network is a minimum. The network was tested using the test data generated and the normalised error on test calculated. The experiment was run for 100 different random start configurations for the weights. The solution for the weights which gave the lowest normalised error on on the training set over the 100 experiments was chosen as the one which best describes the mapping from image space to position space for the particular network under consideration. This solution is the one used in the analysis of the performance of the network in Section 5.5.

5.4 Maximum Likelihood Solution

Before we give results for the radial basis function and the multilayer perceptron network estimators of position, we consider a maximum likelihood approach to position estimation. It is shown in Appendix A that the maximum likelihood estimate of position is that value of $\boldsymbol{\theta}$ for which the quadratic form, Q, given by

$$Q = \frac{|\boldsymbol{h}(\boldsymbol{\theta})^* \boldsymbol{N}^{-1} \boldsymbol{I}_n|^2}{\boldsymbol{h}(\boldsymbol{\theta})^* \boldsymbol{N}^{-1} \boldsymbol{h}(\boldsymbol{\theta})} \tag{34}$$

is a maximum. In principal, the maximum of Q may be found using some nonlinear optimisation strategy. However, since in general we do not know the function, $\boldsymbol{h}(\boldsymbol{\theta})$ continuously, but only at a finite set of points determined by a calibration procedure and given as the training set, then the value of the quadratic form can only be evaluated at these positions. For the training set considered in this paper, these data points are equally spaced in position. One estimate of source position would be to take the position at which Q is greatest. This would give an estimate of position to an accuracy determined by the sample spacing. A more accurate estimate of position would be to interpolate the sample values and adopt the position of the peak of the interpolating function as the estimate of position. This was the procedure adopted in [27].

A maximum likelihood method has been implemented for the 5×1 array, with a data set consisting of a set of images of sources at equally-spaced positions. The set of data vectors contains 101 images of dimension 5, together with associated positions, equally spaced across the field of view from -2.5 to 2.5 (*i.e.* a spacing of $\frac{1}{20}$). The procedure for determining the maximum likelihood estimate of position given the image of a single source corrupted by noise is

1. Calculate the value of the quadratic form (34) at each point of the training set.
2. Find the peak value.
3. Fit a quadratic function to the 3 data points centred on the peak position.
4. Select the position of the source as the peak of the interpolating quadratic function.

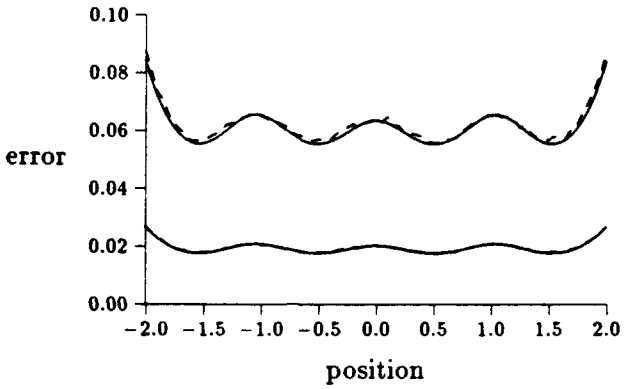

Figure 4: The root of the total square error as a function of position for the linear array and the maximum likelihood method for a value of σ^2 of 10^{-2} (upper curves) and 10^{-3} (lower curves)

Figure 4 plot the root mean square error in position as a function of position for the maximum likelihood method for values of σ^2 of 10^{-2} and 10^{-3}. The solid lines are the analytic approximations derived in Appendix A.

$$\langle \epsilon_\theta^2 \rangle^{\frac{1}{2}} = \frac{\Theta_B}{K_\theta \sqrt{snr_{ref}}} \tag{35}$$

where Θ_B is the beamwidth (unity in this example) and K_θ is a function of θ. The dashed lines are the result of Monte–Carlo simulations based on 5000 images at each position. The estimate of position was made using the method described above.

For both values of signal–to–noise ratio, there is very good agreement between the results obtained using the Monte–Carlo simulation and the high signal–to–noise theoretical predictions. At lower signal–to–noise ratios, we would expect deviation between the simulation and the theory to increase since the analytic approximation for the error derived in the appendix was derived for a high signal–to–noise ration régime. Also, at very high signal–to–noise ratios, there would be deviation between theory and experiment. This is because there is a limit on the error (even in the absence of noise) imposed by the approximate nature of the maximum likelihood solution, which is based on a finite number of samples of the point–spread function and a quadratic interpolation to the quadratic form, Q. We have found that the bias introduced by sampling the point–spread function and quadratic interpolation is less than 7.0×10^{-4} over the central region of the field of view. This is much smaller than the noise errors for the values of signal–to–noise ratio used in the illustrations and only becomes similar to the noise error at signal–to–noise ratios of about 10^6.

5.5 Feed–forward Network Results

Figures 5 and 6 plot the variance in the radial basis function approximator for several values of σ^2, again obtained using a Monte–Carlo simulation. Figure 5 plots the variance over the field of view for the same values of σ^2 used to illustrate the maximum likelihood estimator.

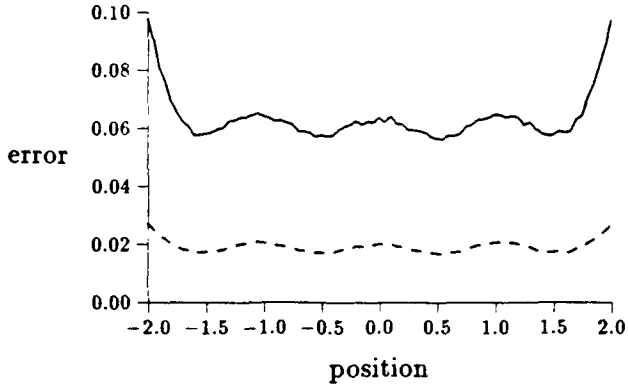

Figure 5: The root of the total square error as a function of position for the linear array and a Radial Basis Function network trained with $\sigma^2 = 10^{-2}$ (solid line) and $\sigma^2 = 10^{-3}$ (dashed line)

The performance is very similar. Figure 6 is a *zoomed-in* plot of the central region of the field of view for three values of σ^2. It shows that for $\sigma^2 = 10^{-4}$, the network is unable to interpolate between points in the training set - hence the saw-tooth effect of the error. It peaks at a value of 0.025 (which is half of the sample spacing in the data set). The reason for this failure is that the approximation given by Equation (30) is not valid since the varaince in the noise is smaller than the sample spacing of data vectors.

Therefore, in order to achieve estimates of position more accurate than that permitted by the spacing of points in the training set, a parametric form must be adopted for the approximating function. The parameters of this function may then be obtained by a suitable optimisation strategy which minimises an error between the approximation and the desired values.

Multilayer perceptron results are given in Figures 7 - 13. Several multilayer perceptron networks, each with a single hidden layer with a different number of hidden units, were trained and the normalised error[1] on the train and test sets calculated. In the first instance, the networks were trained to minimise the sum-squared error between the actual output of the network and the desired output for the training set. Figure 7 plots the normalised errors as a function of the number of hidden units. A normalised error of 10^{-2} on the test set corresponds to a root mean sum-squared error in position of 1.15×10^{-2} of a beamwidth and a normalised error of 10^{-4} on the test set corresponds to a root mean sum-squared error in position of 1.15×10^{-4}. The figure shows that the training error is a monotonic decreasing function of the number of hidden units, whilst the test error decreases up to 5 hidden units and then begins to fluctuate. Therefore, we have selected a network with 5 hidden units to illustrate the results. Figure 8 plots the bias in position (the difference between the actual position and the position predicted using the network) as a function of actual source position for a trained network with 5 hidden units over the test interval, $[-2.0, 2.0]$. The normalised error on the test set is 2.09×10^{-4} and corresponds to a root of the mean sum-squared error in position of 2.4×10^{-4} of a beamwidth and the peak value

[1] The normalised error is the square root of the ratio of the mean sum-squared error to the variance in the target values [30]

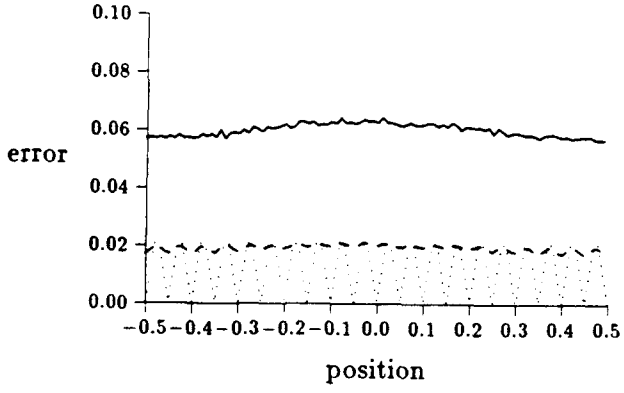

Figure 6: The root of the total square error as a function of position for the linear array and a Radial Basis Function network trained with $\sigma^2 = 10^{-2}$ (solid line), $\sigma^2 = 10^{-3}$ (dashed line) and $\sigma^2 = 10^{-4}$ (dotted line)

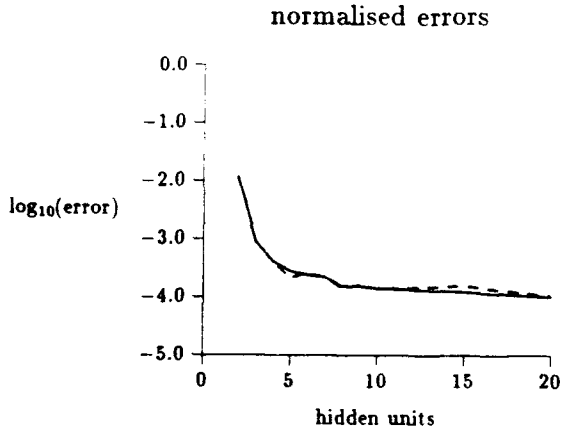

Figure 7: $\mathbf{Log_{10}}$**(normalised error) on the training set (solid line) and the test set (dashed line) for the linear array.**

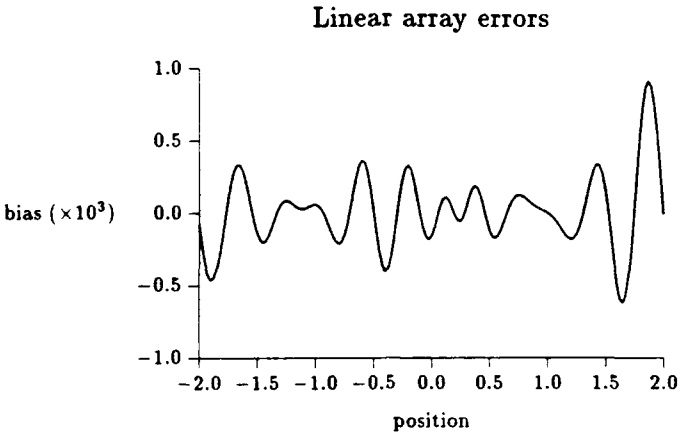

Figure 8: Bias ($\times 1000$) as a function of position for a linear array and a network with 5 hidden units, trained on 101 patterns.

of the bias error is 9.0×10^{-4} of a beamwidth. These errors are very small and therefore, from the experiments with the linear data, we conclude that it is possible to achieve a very accurate nonlinear mapping from the image vector to position. However, there will be errors in the position estimate due to noise on the inputs. Figure 9 plots the standard deviation in the position estimate, $\langle(\theta-\theta_0)^2\rangle^{\frac{1}{2}}$, obtained using a Monte–Carlo simulation for noise on the inputs of value $\sigma^2 = 10^{-3}$. It is immediately apparent that this is significantly greater than

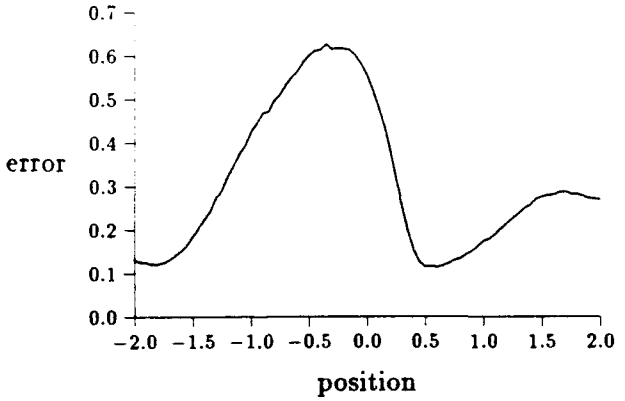

Figure 9: The root of the total square error as a function of position for the linear array and a network with 5 hidden units for noise on the inputs with value $\sigma^2 = 10^{-3}$

the maximum likelihood method or the radial basis function approximation. The reason for this is that the network has been trained to minimise the sum-squared error on a training set which is not representative of the data used to test the network (*i.e.* the training set is noiseless) and has not been trained for the operating conditions of noisy images.

The effects of training on noisy vectors (*i.e.* data representative of the expected operating

conditions) may be simulated by training on the noiseless data but modifying the error criterion used for training. In the final experiments illustrated here, a multilayer perceptron was trained to minimise the augmented error given by Equation (15). A value of 10^{-3} was taken for σ^2 and the results are given in Figures 10 and 11. Figure 10 plots the bias as

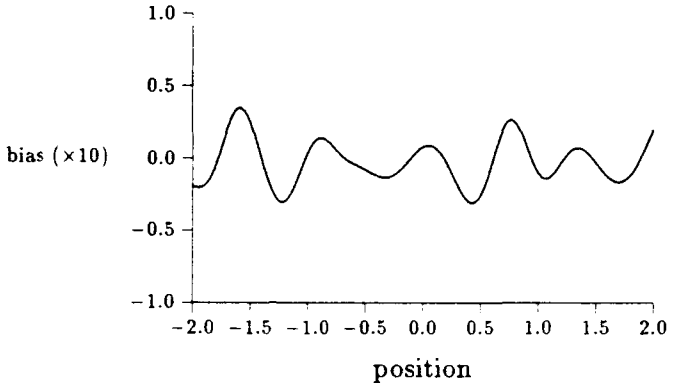

Figure 10: Bias (×10) as a function of position for a linear array and a network with 5 hidden units, trained on 101 patterns, and with a value of σ^2 of 10^{-3}.

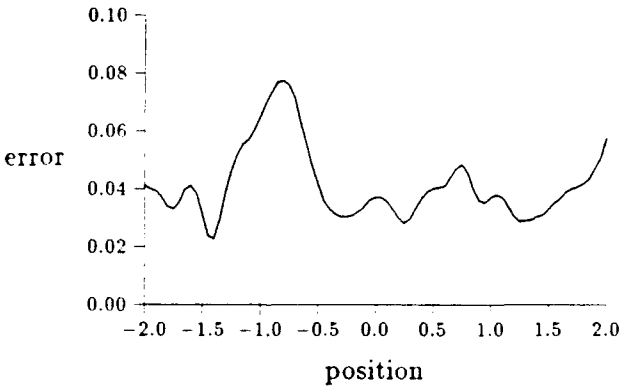

Figure 11: The root of the total square error as a function of position for the linear array and a network with 5 hidden units

a function of position. This is considerably greater than that shown in Figure 8, but the root of the total error (calculated using a Monte–Carlo simulation with input noise of 10^{-3} and given in Figure 11) is reduced compared to Figure 9. Thus, it is possible to reduce the total squared error in position for a multilayer perceptron operating on noisy data by taking into account the expected operating conditions during the training procedure. The errors are still not so small as the errors given by the maximum likelihood method or the radial basis function network, but it can be reduced further by the addition of more hidden units. A multilayer perceptron with 25 hidden units reduces the total square error (averaged across the test set) from 1.78×10^{-3} obtained for 5 hidden units to 5.04×10^{-4}. Results for 25 hidden units are given in Figures 12 and 13. Also, since the multilayer perceptron

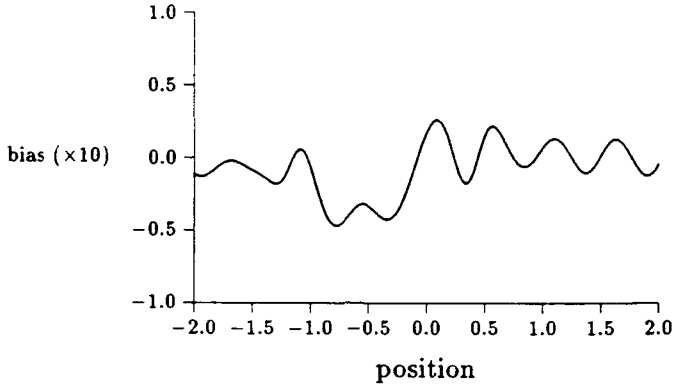

Figure 12: Bias (×10) as a function of position for a linear array and a network with 25 hidden units, trained on 101 patterns, and with a value of σ^2 of 10^{-3}.

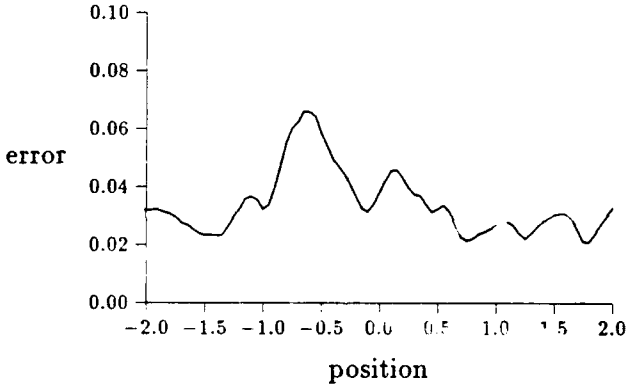

Figure 13: The root of the total square error as a function of position for the linear array and a network with 25 hidden units

is making a global fit to the data, it is not so sensitive to the sample spacing as the radial basis function network. Of course, increasing the number of hidden units (and hence the number of free parameters to adjust) may lead to overfitting of the data.

We conclude this section with a short discussion of the three methods which we have considered in this paper. The main points are summarised in Table 1. Firstly, both the maximum likelihood method and the radial basis function network require storage of the point-spread function; that is, all the training data is required for implementation of the methods. For the problem considered in this paper, or indeed even for the two-dimensional array, the amount of data is not excessive. However, this may not be so in problems where the input and output dimensions are large. The multilayer perceptron, on the other hand, parameterises the point-spread function in the weights of the of the network[2].

[2]Of course, a radial basis function network could be constructed with a reduced number of centres [3] since it is not necessary to have at centre at every data point. However, in these comparisons, we consider

	METHOD		
property	RBF	MLP	MLP
Storage requirements	all training data	weights	all training data
global or local method	local	global	local
SNR régime	low SNR	high SNR	any value
SNR dependence	requires σ^2	requires σ^2	does not require σ^2

Table 1: Summary of properties of the methods discussed in this paper.

The multilayer perceptron performs a global fit to the training data. The maximum likelihood method is a local method in that, once the position corresponding to the peak of the quadratic form is determined, only local points are used to obtain a more accurate estimate of position. The radial basis function network uses all data points to estimate the position of the source, but the contribution from those which are distant from the input vector is minimal so that it is effectively a local method.

Both the multilayer perceptron and the radial basis function network require knowledge of the signal-to-noise ratio. A different value of σ^2 requires a different network. Thus, a network must be constructed for each different signal-to-noise ratio régime or some means of adapting the weights of the multilayer perceptron or the nonlinear functions in the radial basis function network must be employed. The maximum likelihood method does not require a knowledge of σ^2. For a given image vector, the position of the maximum of the quadratic form Q (see Equation (34)) is independent of σ^2.

The particular radial basis function network aproximation derived in this paper is valid for low signal-to-noise ratios. The multilayer perceptron has been derived for high signal-to-noise ratios, but could be extended to lower signal-to-noise ratios by including higher order terms in the expansion of the error. The maximum likelihood method is appropriate for any value of σ^2, though the analytic expressions for the bias and variance in the estimate, derived in the appendix, are valid for a high signal-to-noise approximation.

6 Discussion.

This paper has considered a functional approximation approach to point-source location using an array of sensors. Specifically, the array of sensors considered was a focal-plane array radar and the position of a single source in the scene giving rise to a measured image was regarded as a nonlinear function of that image. The problem then is, given some training data comprising representative images of point sources and their associated positions, define a mapping from image space to position which is robust to noise on the image. A minimum

the particular radial basis function network which arises as a consequence of approximating the integral (28) by a finite sum over the training set.

mean square error approach was adopted since this gives an approximant which is the expected value of the position for a given image.

The approximant which gives the expected value of the *a posteriori* density may be expressed as a sum over the training set giving the form of a radial basis function network. This is valid in the situation where the noise variance is large compared to the sample spacing of data points in the training set. In the other extreme of small noise, we must assume some parametric form for the approximant and we adopted a multilayer perceptron architecture. We evaluated the performance of both of these feed-forward network architectures and compared them with a maximum likelihood approach.

For a low signal-to-noise ratio, the errors in the position estimate for the radial basis function network and the maximum likelihood approach are very similar. At higher signal-to-noise ratios, the radial basis function approximation becomes increasingly invalid and a prescribed parametric form (the multilayer perceptron) was used. This was trained using an augmented error criterion to simulate the effects of noise on the expected data 'in operation'. A MLP with 5 hidden units did not perform so well as the maximum likelihood method. Increasing the number of hidden units to 25 improved the performance, but the maximum likelihood method was still superior. A further advantage of the maximum likelihood method is that it does not depend on the noise power, σ^2, whereas the RBF and the MLP approximants are functions of σ^2.

One advantage of exploring the MLP architecture is that it is general purpose and there is the potential for implementation on the focal plane of the array, which may give significant data reduction on the array and which may be very important in some applications. The approach of regarding the position of a point source as a nonlinear function of the image also has application to staring array sensors other than radar in which it is required to obtain sub-pixel accuracy of a source (eg [5]). Obviously, the work can be extended to two dimensional arrays (see [27] for the maximum likelihood method applied to square and hexagonal two-dimensional arrays) but the study in this paper was restricted to one dimension for illustration purposes.

There are several possible avenues for further work. Improved performance may be obtained for the MLP if the nonlinear functions at the hidden nodes were better matched to this particular problem. Also, it may be appropriate for some applications if the estimate of the position were unbiased so that integration may take place after position estimation. Application to real radar data will require some modification to the MLP architecture, since the data vectors will be complex, and the MLP must be designed so that it is insensitive to an arbitrary phase associated with the target. This is not a difficult problem. Further, can the functional approximation approach be applied to multi-source scenes? A direct implementation of the method would lead to a vast amount of training data covering all possible positions and relative amplitudes of sources. Therefore, some other architecture may be more appropriate (see [9, 12] for an approach based on Hopfield networks). However, the single source assumption is valid where range and doppler processing can be employed to discriminate between sources and, after all, is the assumption which monopulse radar makes.

In conclusion, a novel approach to point-source location based on function approximation has been presented and compared with a more traditional solution. The potential for implementation of the method on the focal-plane of the sensor could be significant.

Appendix A Maximum Likelihood Solution

The conditional probability of an observation, $I_n \in \mathbb{R}^m$, given the position, θ, and amplitude, A, of a source for a Gaussian noise process is given by

$$p(I_n|\theta, A) = \frac{1}{(2\pi)^{m/2}|N|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(I_n - Ah(\theta))^* N^{-1}(I_n - Ah(\theta))\right) \quad (36)$$

where N, is the $m \times m$ positive semi-definite covariance matrix of additive noise.

Thus, the log likelihood, $log(p(I_n|\theta, A))$ is given by

$$log\,(p(I_n|\theta, A)) = -\frac{m}{2}\,log(2\pi) - \frac{1}{2}log(|N|) - \frac{1}{2}(I_n - I_\theta)^* N^{-1}(I_n - I_\theta), \quad (37)$$

where

$$I_\theta = Ah(\theta), \quad (38)$$

is the image of a source of amplitude A at position θ in the absence of noise.

Since the first two terms in Equation (37)are independent of the parameters A and θ, the maximum of the likelihood function occurs when the quadratic form

$$(I_n - I_\theta)^* N^{-1}(I_n - I_\theta), \quad (39)$$

is a minimum.

Differentiating the above expression with respect to the parameter A and equating to zero gives the maximum likelihood solution for A, expressed in terms of $h(\theta)$ as

$$A = \frac{h^*(\theta)N^{-1}I_n}{h^*(\theta)N^{-1}h(\theta)} \quad (40)$$

Substituting for A into the expression (39) and simplifying the algebra, we find that the expression is now a function of θ alone (through $h(\theta)$) and that a minimum occurs when the quantity E, given by

$$E = I_n^* N^{-1} I_n - \frac{|h^* N^{-1} I_n|^2}{h^* N^{-1} h} \quad (41)$$

is a minimum. Since the first term is independent of θ, the maximum likelihood solution for θ occurs when the second quantity (including the minus sign) is a minimum, i.e. when

$$\frac{|h^* N^{-1} I_n|^2}{h^* N^{-1} h} \quad (42)$$

is a maximum.

For $N = \sigma^2 I$, the quadratic form reduces to

$$\frac{1}{\sigma^2}\frac{|h^* I_n|^2}{h^* h} \equiv \frac{1}{\sigma^2}|\hat{h}^* I_n|^2 \quad (43)$$

where $\hat{\boldsymbol{h}}$ is the normalised point-spread function vector (normalised to unit magnitude). This is a function of $\boldsymbol{\theta}$ alone, and the maximum likelihood estimate of position is the value of $\boldsymbol{\theta}$ at which the above quantity attains its maximum. In order to determine this value some means of nonlinear optimisation must be employed since, in general, it may not be possible to write down a solution in closed form.

However, we can obtain expressions for the bias and the variance in the maximum likelihood estimator (at least in a small noise approximation) from a perturbation expansion as follows. Differentiating the expression (41) with respect to $\boldsymbol{\theta}$, and equating to zero give.

$$(\boldsymbol{h}^* \boldsymbol{N}^{-1} \boldsymbol{h}) \left(\boldsymbol{I}_n^* \boldsymbol{N}^{-1} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \right) - (\boldsymbol{I}_n^* \boldsymbol{N}^{-1} \boldsymbol{h}) \left(\boldsymbol{h}^* \boldsymbol{N}^{-1} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \right) = 0 \tag{44}$$

In the absence of noise, the solution is given by $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. When noise is present, let the image be given by

$$\boldsymbol{I}_n = A_0 \boldsymbol{h}(\boldsymbol{\theta}_0) + \boldsymbol{n} \tag{45}$$

where A_0 and $\boldsymbol{\theta}_0$ are the true values of amplitude and position and $\boldsymbol{n}$ is the perturbation of the image due to noise.

For a small perturbation to the noiseless image given by the noise vector, $\boldsymbol{n}$, let the solution for $\boldsymbol{\theta}$ be $\boldsymbol{\theta}_0 + \epsilon_{\boldsymbol{\theta}}$. Substituting this into Equation (43) and expanding the functions $\boldsymbol{h}(\boldsymbol{\theta})$ and $\partial \boldsymbol{h} / \partial \boldsymbol{\theta}$ using Taylor's theorem leads to solutions for the mean and the standard deviation[3] of the estimate as

$$\langle \epsilon_{\boldsymbol{\theta}} \rangle = \frac{\sigma^2}{A_0^2 \alpha^2} \left\{ \boldsymbol{h}^* \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \alpha - \frac{5}{2} \boldsymbol{h}^* \boldsymbol{h} \beta \right\} \tag{46}$$

and

$$\langle \epsilon_{\boldsymbol{\theta}}^2 \rangle^{\frac{1}{2}} = \left(\frac{\boldsymbol{h}^* \boldsymbol{h}}{\alpha} \frac{\sigma^2}{A_0^2} \right)^{\frac{1}{2}} \tag{47}$$

where

$$\alpha = -\boldsymbol{h}^* \boldsymbol{h} \frac{\partial \boldsymbol{h}^*}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} - \left(\frac{\partial \boldsymbol{h}^*}{\partial \boldsymbol{\theta}} \right)^2 \tag{48}$$

and

$$\beta = \frac{\partial^2 \boldsymbol{h}^*}{\partial \boldsymbol{\theta}^2} \boldsymbol{h} \frac{\partial \boldsymbol{h}^*}{\partial \boldsymbol{\theta}} \boldsymbol{h} - \boldsymbol{h}^* \boldsymbol{h} \frac{\partial^2 \boldsymbol{h}^*}{\partial \boldsymbol{\theta}^2} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \tag{49}$$

Defining the signal-to-noise ratio to be the total power received by the array of sensors for a source in a reference position (usually taken to be the centre of the field of view) divided by the noise power per receiver[4]

$$snr_{ref} = \frac{|A_0|^2}{\sigma^2} \boldsymbol{h}^*(\boldsymbol{\theta}_{ref}) \boldsymbol{h}(\boldsymbol{\theta}_{ref}) \tag{50}$$

[3] The standard deviation may also be obtained by using the result that the maximum likelihood estimate is asymptotically normally distributed with a dispersion matrix depending on the likelihood function [14]. This was the approach considered in [27]

[4] This gives a position independent definition of signal-to-noise ratio

then the standard deviation may be written in the form

$$\langle \epsilon_\theta^2 \rangle^{\frac{1}{2}} = \frac{\Theta_B}{K_\theta \sqrt{snr_{ref}}} \tag{51}$$

where

$$K_\theta = \left(\frac{\alpha}{h^*(\theta)h(\theta)h^*(\theta_{ref})h(\theta_{ref})} \right)^{\frac{1}{2}} \Theta_B \tag{52}$$

and Θ_b is the beamwidth. This is a form often quoted for the tracking error in monopulse radar. It shows that the standard deviation is inversely proportional to the square root of the signal-to-noise ratio, with the constant of proportionality being a function of position. Figure 14 plots the quantity K_θ as a function of position.

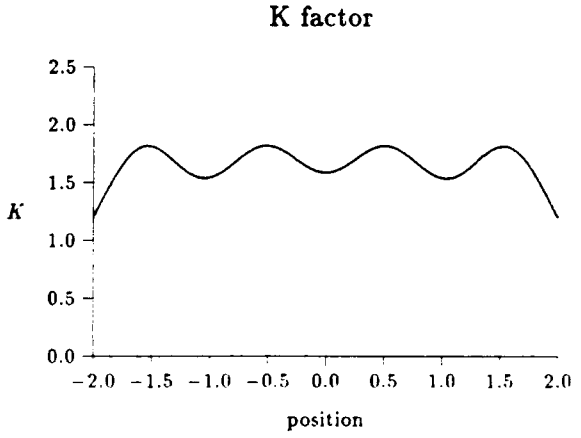

Figure 14: The quantity K as a function of position for the linear array.

Similarly, the mean of the estimate may be written

$$\langle \epsilon_\theta \rangle = \frac{\Theta_B B_\theta}{snr_{ref}} \tag{53}$$

where

$$B_\theta = \frac{1}{\alpha^2} \left(h \frac{\partial h}{\partial \theta} \alpha - \frac{5}{2} h^* h \beta \right) h^*(\theta_{ref}) h(\theta_{ref}) \tag{54}$$

This shows that the mean is inversely proportional to the signal-to-noise ratio. Figure 15 plots the quantity B_θ as a function of position.

These results show that, at high signal-to-noise ratio, the bias is small compared to the standard deviation.

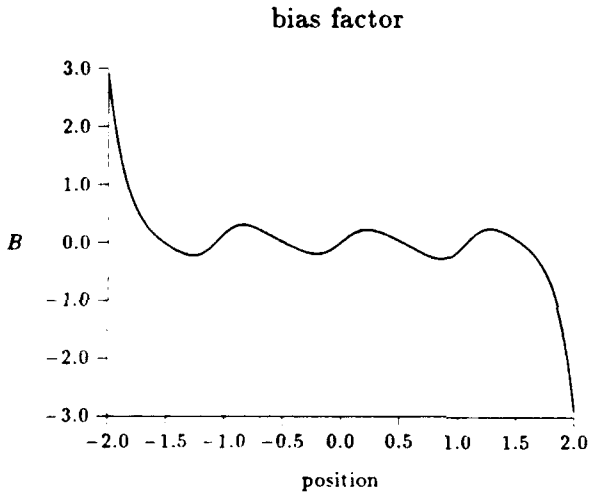

Figure 15: The bias factor, B, for a linear array

Appendix B Expected Error

Let the pattern $\boldsymbol{x}_p$ be corrupted by additive noise, $\boldsymbol{n}$, so that the error for pattern $\boldsymbol{x}_p$ is

$$E_p = E(\boldsymbol{x}_p + \boldsymbol{n}) = E(\boldsymbol{x}_p) + (\boldsymbol{n}^*\nabla)E|_{\boldsymbol{x}_p} + (\boldsymbol{n}^*\nabla)^2 E\Big|_{\boldsymbol{x}_p} \tag{55}$$

expanding by Taylor's theorem and assuming that $\boldsymbol{n}$ is small so that terms $\mathcal{O}(|\boldsymbol{n}|^3)$ may be neglected. For $\langle \boldsymbol{n} \rangle = 0$, the expected error (average over all noise vectors) is

$$\langle E_p \rangle = E(\boldsymbol{x}_p) + \frac{1}{2}\langle \boldsymbol{n}^* \boldsymbol{H}^p \boldsymbol{n} \rangle \tag{56}$$

where $E(\boldsymbol{x}_p)$ is the error in the absence of noise and $\frac{1}{2}\langle \boldsymbol{n}^* \boldsymbol{H}^p \boldsymbol{n} \rangle$ is an additional error term where $\boldsymbol{H}^p$ is the Hessian with respect to the data space components, evaluated for the pth pattern

$$\boldsymbol{H}^p_{ij} = \frac{\partial^2 E}{\partial x_i \partial x_j}\Big|_{\boldsymbol{x}_p}. \tag{57}$$

For $\langle n_i n_j \rangle = \sigma^2 \delta_{ij}$, the additional error term may be written

$$\frac{1}{2}\langle \boldsymbol{n}^* \boldsymbol{H}^p \boldsymbol{n} \rangle = \frac{\sigma^2}{2} Tr(\boldsymbol{H}^p), \tag{58}$$

where Tr is the matrix trace operation. Averaging over all data patterns gives the mean expected error

$$\langle E_T \rangle = \frac{1}{P}\sum_{p=1}^{P} E(\boldsymbol{x}_p) + \frac{\sigma^2}{2P}\sum_{p=1}^{P} Tr(\boldsymbol{H}^p). \tag{59}$$

References

[1] C.J. Alder, C.R. Brewitt-Taylor, M. Dixon, R.D. Hodges, L.D. Irving, and H.D. Rees. Lens-fed Microwave and Millimetre-wave Receivers with Integral Antennas. In 20^{th} *European Microwave Conference*, Budapest, 1990.

[2] W.D. Beastall. Recognition of Radar Signals by Neural Network. In *First IEE Int. Conf. on Artificial Neural Networks*, pages 139–142, London, October 1989. IEE Conf. Publ. 313.

[3] D.S. Broomhead and D. Lowe. Multi-variable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2(3):269–303, 1988.

[4] P.F. Castelaz. Neural Networks in Defense Applications. In *IEEE Int. Conf. Neural Networks*, volume II, pages 473–480, San Diego, 1988.

[5] J.A. Cox. Point-source Location Using Hexagonal Detector Arrays. *Optical Engineering*, 26(1):69–74, January 1987.

[6] S.E. Decatur. Application of Neural Networks to Terrain Classification. In *IEEE Int. Joint Conf. Neural Networks*, volume I, pages 283–288, Washington D.C., 1989.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Inc., London, 1972.

[8] R.P. Gorman and T.J. Sejnowski. Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. *Neural Networks*, 1(1):75–90, 1989.

[9] D. Goryn and M. Kaveh. Neural Networks for Narrowband and Wideband Direction Finding. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages 2164–2167, New York, USA, 1988. IEEE.

[10] K. Hornik, M. Stinchcombe, and H. White. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2(5):359–366, 1989.

[11] T.M. Jelonek and J.P. Reilly. Maximum Likelihood Estimation for Direction of Arrival Using a Nonlinear Optimising Neural Network. In *IEEE Int. Joint Conf. Neural Networks*, volume I, pages 253–258, San Diego, CA, 1990.

[12] S. Jha, R. Chapman, and T.S. Durrani. Bearing Estimation Using Neural Networks. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages 2156–2159, New York, USA, 1988. IEEE.

[13] S.K. Jha and T.S. Durrani. Bearing Estimation Using Neural Optimisation Methods. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 2, pages 889–892, Albuquerque, USA, 1990. IEEE.

[14] M.G. Kendall and A. Stuart. *The Advanced Theory of Statistics*, volume 2. Charles Griffin and Company Limited, London, second edition, 1967.

[15] R.M. Kuczewski. Neural Network Approaches to Multi-target Tracking. In *IEEE First Int. Conf. Neural Networks*, volume IV, pages 619–633, San Diego, 1987.

[16] A. Lapedes and R. Farber. How Neural Nets Work. In D.Z. Anderson, editor, *Neural Information Processing Systems*, pages 442–456. AIP, Denver, 1987.

[17] D. Lowe and A.R. Webb. Optimised Feature Extraction and the Bayes Decision in Feed-forward Classifier Networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 4(13):355–364, April 1991.

[18] J.L. Mather. Least Squares Solutions in Signal Processing Using the Singular Value Decomposition. RSRE Memo 3864, Royal Signals and Radar Establishment, R.S.R.E., St Andrews Road, Malvern, Worcs., WR14 3PS, 1986.

[19] S.M. Peeling and R.K. Moore. Isolated Digit Recognition Experiments Using the Multilayer Perceptron. *Speech Communication*, 7:403–409, 1988.

[20] P.A. Penz, A. Katz, M.T. Gately, D.R. Collins, and J.A. Anderson. Analog Capabilities of the BSB Model as Applied to the Anti-radiation Homing Missile Problem. In *IEEE Int. Joint Conf. Neural Networks*, volume II, pages 7–12, Washington D.C., 1989.

[21] M.W. Roth. Survey of Neural Network Technology for Automatic Target Recognition. *IEE Trans. Neural Networks*, 1(1):28–43, March 1990.

[22] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learing Internal Representations by Error Propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing : Exploration in the Microstructure of Cognition*, pages 318–362. Cambridge : MIT Press, 1986.

[23] R.O. Schmidt. Multiple Emitter Location and Signal Parameter Estimation. In *Proceedings of the RADC Spectrum Estimation Workshop*, pages 243–258, 1979.

[24] M. Stinchcombe and H. White. Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights. In *IEEE Int. Joint Conf. Neural Networks*, volume III, pages 7–16, San Diego, 1990.

[25] A.R. Webb. Applications of Neural Networks in Military Systems. In *Military Microwaves '90*, pages 356–361, London, 1990. Microwave Exhibitions and Publishers Ltd.

[26] A.R. Webb. Functional Approximation in Feed-forward Networks: A Least-squares Approach to Generalisation. RSRE Memo 4453, R.S.R.E., St Andrews Road, Malvern, Worcs., WR14 3PS, 1990.

[27] A.R. Webb. Point-source Location Using a Millimetre-wave Focal-plane Array Radar. *IEE Proceedings Part F*, 1991. to appear.

[28] A.R. Webb and D. Lowe. A Hybrid Optimisation Strategy for Adaptive Feed-forward Layered Networks. RSRE Memo 4193, R.S.R.E., St Andrews Road, Malvern, Worcs., WR14 3PS, 1988.

[29] A.R. Webb and D. Lowe. The Optimised Internal Representation of Multilayer Classifier Networks Performs Nonlinear Discriminant Analysis. *Neural Networks*, 3(4):367–375, July/August 1990.

[30] A.R. Webb, D. Lowe, and M.D. Bedworth. A Comparison of Nonlinear Optimisation Strategies for Feed-forward Adaptive Layered Networks. RSRE Memo 4157, R.S.R.E., St Andrews Road, Malvern, Worcs., WR14 3PS, 1988.

[31] A. Wieland and R. Leighton. Geometric Analysis of Neural Network Capabilities. In *IEEE First Int. Conf. Neural Networks*, volume III, pages 385–392, San Diego, 1987.

REPORT DOCUMENTATION PAGE

DRIC Reference Number (if known) .....

Overall security classification of sheet .....UNCLASSIFIED.....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the field concerned must be marked to indicate the classification eg (R), (C) or (S).

Originators Reference/Report No.	Month	Year
MEMO 4567	APRIL	1991

Originators Name and Location
RSRE, St Andrews Road
Malvern, Worcs WR14 3PS

Monitoring Agency Name and Location

Title

A COMPARISON OF FEED-FORWARD NETWORKS AND MAXIMUM LIKELIHOOD ON A POINT-SOURCE LOCATION PROBLEM

Report Security Classification	Title Classification (U, R, C or S)
UNCLASSIFIED	U

Foreign Language Title (in the case of translations)

Conference Details

Agency Reference	Contract Number and Period

Project Number	Other References

Authors	Pagination and Ref
WEBB, A R	30

Abstract

The problem of point source location using a multi-beam focal-plane staring array radar is addressed. It is viewed as one in functional approximation in which the position of the source is regarded as a nonlinear function of the sampled radar image and it is required to construct an approximant, using a training set, which minimises the mean square error in the position estimate. The problem is also one of generalisation, since the expected operating conditions are likely to be corrupted by noise and this must be taken into account when designing the approximant. Two feed-forward network architectures are considered - a particular radial basis function network which arises as a consequence of the minimum mean square error solution and is appropriate when the signal-to-noise ratio is 'small' and a multi-layer perceptron, chosen for high signal-to-noise ratio approximation. The errors in the position estimates for each of these approaches are compared with a maximum likelihood position estimation method. The maximum likelihood method gives better overall performance and has the advantage that it is not dependent on the signal-to-noise ratio.

Abstract Classification (U,R,C or S)
U

Descriptors

Distribution Statement (Enter any limitations on the distribution of the document)

UNLIMITED

sssss

INTENTIONALLY BLANK