Nonlinear processing of non-Gaussian stochastic and chaotic deterministic time series

Mark Cowper

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
March 2000

Abstract

It is often assumed that interference or noise signals are Gaussian stochastic processes. Gaussian noise models are appealing as they usually result in noise suppression algorithms that are simple: i.e. linear and closed form. However, such linear techniques may be sub-optimal when the noise process is either a non-Gaussian stochastic process or a chaotic deterministic process. In the event of encountering such noise processes, improvements in noise suppression, relative to the performance of linear methods, may be achievable using nonlinear signal processing techniques. The application of interest for this thesis is maritime surveillance radar, where the main source of interference, termed sea clutter, is widely accepted to be a non-Gaussian stochastic process at high resolutions and/or at low grazing angles. However, evidence has been presented during the last decade which suggests that sea clutter may be better modelled as a chaotic deterministic process. While the debate over which model is more suitable continues, this thesis investigates whether nonlinear processing techniques can be used to improve the performance of maritime surveillance radar, relative to the performance achievable using linear techniques.

Linear and nonlinear prediction of chaotic signals, sea clutter data sets, and stochastic surrogate clutter data sets is carried out. Volterra series filter networks and radial basis function networks are used to implement nonlinear predictors. A novel structure for a forward-backward nonlinear predictor, using a radial basis function network, is presented. Prediction results provide evidence to support the view that sea clutter is better modelled as a stochastic process, rather than as a chaotic process. The clutter data sets are shown to have linear predictor functions. Linear and nonlinear predictors are used as the basis of target detection algorithms. The performance of these predictor-detectors, against backgrounds of sea clutter data and against a background of chaotic noise data is evaluated. The detection results show that linear predictor-detectors perform as well as, or better than, nonlinear predictor-detectors against the non-Gaussian clutter backgrounds considered in this thesis, whilst the reverse is true for a background of chaotic noise.

An existing, nonlinear inverse, noise cancellation technique, referred to as Broomhead's filtering technique in this thesis, is re-investigated using a sine wave corrupted by broadband chaotic noise. It is demonstrated that significant improvements can be obtained using this nonlinear inverse technique, relative to results obtained using linear alternatives, despite recent work which suggested otherwise. A novel bandstop filtering approach is applied to Broomhead's filtering method, which allows the technique to be applied to the cancellation of signals with a band of interest greater than that of a sine wave. This modified Broomhead filtering technique is shown to cancel broadband chaotic noise from a narrowband Gaussian signal better than alternative linear methods. The modified Broomhead filtering technique is shown to only perform as well as, or more poorly than, a linear technique on narrowband Gaussian signals corrupted by clutter data.

“To strive, to seek, to find, and not to yield.”

- *Taken from Ulysses by Alfred Lord Tennyson*

Declaration

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the Department of Electronics and Electrical Engineering at the University of Edinburgh.

Mark Cowper
March 2000

Acknowledgements

This is the part where I pay my thanks to those who have helped me at some stage or another during my PhD. If you are the type of person who cannot stand such things as Oscar acceptance speeches, please turn to the next page now!

Thanks to:

- My supervisors Professor Bernie Mulgrew and Doctor Steve McLaughlin for their support and guidance, as well as for their constructive comments on the writing of this thesis.
- Mark Denny from GEC Marconi Avionics for his insightful discussions and emails regarding the modelling of sea clutter and aspects of practical radar target detection.
- GEC Marconi Avionics and EPSRC for providing the financial support for the work reported herein.
- DERA for supplying the sea clutter analysed in this thesis: in particular thanks to Sam Lycett, James Branson and Bill Dawber.
- Charlie Unsworth for providing the software to estimate the mutual information and maximum likelihood correlation dimension of a discrete time series.
- Ian Band for proof-reading parts of this thesis, and for proof-reading conference papers submitted during my PhD studies.
- The members (existing and previous) of the signals and systems research group, particularly John Thompson and George Taylor, who have at some stage or another provided valuable assistance.

Contents

List of figures

List of tables

Acronyms and abbreviations

ACF	autocorrelation function
CDF	cumulative distribution function
DERA	defence evaluation and research agency
FFT	fast Fourier transform
FB	forward-backward
FBLP	forward-backward linear predictor
FB-RBFN	forward-backward radial basis function network
FB-NRBFNP-RSC	forward-backward normalised radial basis function network predictor with centres selected at random from a training data set
HH	horizontal polarisation on transmit and receive
IIR	infinite impulse response
LP	linear predictor
LPD	linear predictor detector
MAF	moving average filter
MIDAS	mobile instrumented data acquisition system
MLP	multilayer perceptron
MNLT	memoryless nonlinear transformation
MPR	multiband pulsed radar
NLP	nonlinear predictor
NLPD	nonlinear predictor detector
NMSE	normalised mean square error
NRBFN	normalised radial basis function network
NRBFNP	normalised radial basis function network predictor
NRBFNP-OAKM	normalised radial basis function network predictor with centres selected using the optimal adaptive k-means clustering algorithm
NRBFNP-RSC	normalised radial basis function network predictor with centres selected at random from a training data set
OAKM	optimal adaptive k-means

pdf	probability density function
PRF	pulse repetition frequency
PSD	power spectral density
RBFN	radial basis function network
RBFNP	radial basis function network predictor
RBFNP-OAKM	radial basis function network predictor with centres selected using the optimal adaptive k-means clustering algorithm
RBFNP-RSC	radial basis function network predictor with centres selected at random from a training data set
RNG	random number generator
RSC	randomly selected centres
RUNRBFN	regularised un-normalised radial basis function network
SCR	signal to clutter ratio
SNR	signal to noise ratio
UNRBFN	un-normalised radial basis function network
UNRBFNP	un-normalised radial basis function network predictor
VS	Volterra series
VSF	Volterra series filter
VSFP	Volterra series filter predictor
VV	vertical polarisation on transmit and receive

List of principal symbols

$x(t)$	observed data
$\mathbf{x}_{Rv}$	reconstruction vector of observed data
D_E	embedding dimension
τ	embedding delay
d	dimension of a chaotic dynamical system
$C(l)$	correlation integral
l	distance measure
l_ε	maximum noise distance
l_0	maximum scaling distance
D_c	correlation dimension
D_{ML}	maximum likelihood correlation dimension
λ_i	i^{th} Lyapunov exponent
D_{KY}	Kaplan-Yorke dimension
λ_w	wavelength of sea waves
λ_r	radar wavelength
ϕ_r	grazing angle
τ_r	radar pulse width
$P_G(.)$	Gaussian pdf
g	Gaussian variate
$P_R(.)$	Rayleigh pdf
$P_e(.)$	exponential pdf
$P_{lN}(.)$	log-Normal pdf
$P_W(.)$	Weibull pdf
$P_K(.)$	K-distribution pdf
ν	K-distribution (and gamma) shape parameter
c	K-distribution scale parameter

$P_c(.)$	chi distribtion pdf
a_c	chi variate
$P_\Gamma(.)$	gamma pdf
z	gamma variate
b	gamma scale parameter
$H(w)$	complex Gaussian correlation filter
ρ	correlation coeffient of IIR correlation filter
$h(.)$	Volterra kernel
$\phi(.)$	Gaussian RBFN kernel function
w_i	i^{th} RBFN output layer weight
$\mathbf{c}_i$	center vector of i^{th} Gaussian RBFN kernel
σ_i	width of i^{th} Gaussian RBFN kernel
$\|\|.\|\|$	Euclidean distance measure
$\hat{x}(n)$	estimate of $x(n)$
λ	regularisation parameter
$\mathbf{I}$	identity matrix
M	number of RBFN kernels
Y	training length
P_{fa}	probability of false alarm
v_t	threshold value
P_d	probability of detection
$t(n)$	target sample at time n
$b(n)$	Broomhead filter input at time n
$b_F(n)$	bandstop filtered Broomhead input at time n
$x_R(n)$	reconstructed Broomhead filter noise process at time n
f/f_s	normalised frequency

Chapter 1
Introduction

1.1 Background

Gaussian stochastic signal models are appealing as they usually result in processing algorithms that are simple: *i.e.* linear and closed form. However, it has been realised that many signals of interest, as well as noise or interference processes, have distinctly non-Gaussian statistics. Indeed, non-Gaussian data are found [1] in a variety of disciplines including astronomy, biology, economics and exploration seismology, to name but a few. Additionally, a tremendous interest in chaos theory from many disciplines [2] during the last 20 years has ignited interest in using chaotic models for data. Chaotic signals are generated from nonlinear deterministic systems and display random-like behaviour [3]. A review of a number of signals that have been analysed for evidence of chaotic behaviour is given in [4]. In the event of encountering non-Gaussian stochastic or chaotic deterministic noise or interference processes, improvements in their suppression, relative to the performance achievable using linear methods, may be gained using nonlinear signal processing techniques. It is the purpose of this thesis to investigate this assertion.

In general, the possible applications of interest for the nonlinear processing techniques discussed in this thesis include situations where the noise or interference can be modelled as a non-Gaussian stochastic or a chaotic deterministic process. The specific application of interest for this thesis is maritime surveillance radar [5, 6], where the main sources of interference are radar reflections from the sea surface, which are termed sea clutter [7]. The radar community widely accepts that sea clutter is well modelled as a non-Gaussian stochastic process at high resolutions and/or at low grazing angles [8]. However, evidence has been presented during the last decade [9–21] which suggests that sea clutter may be better modelled as a chaotic deterministic process. While the debate over which model is more suitable continues, this thesis investigates whether nonlinear signal processing techniques can be used to improve the performance of maritime surveillance radar, relative to the performance achievable using linear methods.

1.2 Thesis structure and original contributions to knowledge

This section summarises the contents of this thesis, as well as highlights the original contributions to knowledge contained within each chapter.

An introduction to what sea clutter is, why it needs to be modelled, and the existing modelling strategies for it are discussed in Chapter 2. The original contributions of this chapter are a thorough review of the evidence for both chaotic and stochastic models of sea clutter, and a review of the criticisms of the techniques used to determine if sea clutter is a chaotic process.

In Chapter 3 the simulation of correlated surrogate sea clutter data is discussed. The stochastic model used for the generation of the surrogate data sets is the compound K-distribution [22]. The original contributions of this chapter are a complete overall methodology for the estimation of the compound K-distributed components of a sea clutter data set, and the simulation of compound K-distributed data with correlated speckle and correlated gamma components. These contributions bring together certain aspects from previous publications on the compound analysis of sea clutter, and from previous publications on the simulation of correlated gamma data.

An investigation into the nonlinear predictability of chaotic signals is presented in Chapter 4. This is done using two nonlinear models: the Volterra series filter (VSF) [23] and the radial basis function network (RBFN) [24]. A linear predictor (LP) is used as a prediction performance benchmark. The original contributions of this chapter are as follows. A detailed investigation into the benefits, if any, of using a clustering algorithm instead of a random subset of the training data for the selection of the kernel centres of a RBFN predictor (RBFNP) are given. Two Gaussian network architectures for a RBFNP are compared. Finally, an investigation into the use of a normalised RBFNP (NRBFNP) to capture the underlying dynamics of a chaotic signal is presented, and compared with the use of un-normalised and regularised RBFNP's to perform the same task.

In Chapter 5 the predictability of sea clutter data sets is discussed using the nonlinear predictor (NLP) structures discussed in Chapter 4. Previous research [9, 12] has shown sea clutter data sets to be nonlinearly predictable. However, all the clutter data sets analysed in [9, 12] for evidence of nonlinear predictability were not very spiky[1]. Additionally, the generalisation

[1] See section 2.3.4 for a description of spiky sea clutter. The significance of spiky data with respect to the predictability of sea clutter is discussed in section 5.2.3.

properties[2] of the NLP's in [9, 12] were not discussed. Furthermore, the comparison of NLP's with LP's for the prediction of clutter in [9, 12] was only carried out on four datasets. The original contributions of this chapter are now given. A rigorous prediction analysis of new[3] sea clutter data sets, which includes spiky data, is presented. The generalisation properties of the predictors are reported and discussed. To investigate the source of the predictability of the clutter data sets, and to determine if a stochastic model is appropriate, prediction analysis is carried out on surrogate, correlated, compound K-distributed data.

In Chapter 6 the predictability of sea clutter is further investigated. The original contributions of this chapter include a prediction analysis of filtered[4] clutter data, with particular attention paid to the generalisation properties of the predictors used. The clutter data is filtered using two different techniques: a smoothing filter and a linear phase FIR lowpass filter. The aim of this filtering is to reduce thermal white noise, whilst leaving the clutter spectra (and therefore any chaotic behaviour) unaffected. A novel structure for a forward-backward RBFNP is presented, and is used to carry out forward-backward prediction on the new sea clutter data sets referred to above.

The applicability of nonlinear predictors to maritime surveillance radar detection is considered in Chapter 7. The original contribution of this chapter is an investigation into what is more important for radar target detection, if clutter were found to be chaotic: normalised mean square error, or capturing the signal's underlying dynamics?

A nonlinear inverse noise cancellation technique [25], referred to as Broomhead's filtering technique in this thesis, is discussed in Chapter 8. This technique uses a normalised RBFN (NRBFN) as an inverse to a linear bandstop filter. The original contributions to knowledge of this chapter are now discussed. A re-investigation of Broomhead's filtering technique for the cancellation of chaotic noise from a sine wave is presented which shows that previously published research [4] which claimed the technique could not be used to perform better, in terms of output signal to noise ratio (SNR), than linear filtering alternatives is incorrect. A novel bandstop filtering approach is applied to Broomhead's filtering technique which allows it to be used to cancel chaotic noise from signals of interest that occupy a larger bandwidth than a sine wave, better than linear cancellation techniques. This modified Broomhead filtering technique

[2] *i.e.* results for non-training data sets.

[3] *i.e.* not previously used in a nonlinear prediction analysis of sea clutter.

[4] Chaotic measures of filtered data (filtered to reduce noise) were reported in [21], however no prediction analysis was reported for the filtered data.

is applied to the cancellation of wideband chaotic noise from narrowband Gaussian signals of interest, and the performance of this method is compared with linear alternatives. The modified Broomhead filtering technique is applied to the cancellation of sea clutter from narrowband Gaussian target signals.

In Chapter 9 a summary of the main points presented in this thesis, conclusions drawn from the work in this thesis, and suggestions for further work to build on that reported herein are presented.

Chapter 2
The modelling of sea clutter

2.1 Introduction

The application of interest for this thesis is maritime surveillance radar, which involves using a radar to search for targets on, or near to, the sea surface. For such an application, the radar reflections from the sea surface, termed sea clutter, are a problem as they can be mistaken for target signals. Understanding the nature of sea clutter is important as such knowledge can be used to prevent sea clutter from degrading the performance of maritime surveillance radars. The accepted view [26] within the radar community is that clutter is best modelled as a stochastic process. However, work conducted during the last decade has provided some evidence to suggest that clutter may be better modelled as a deterministic, chaotic process.

This chapter is structured as follows. In section 2.2 the deterministic modelling of sea clutter is examined in some detail. In section 2.3 the traditional stochastic modelling of sea clutter is discussed. A chapter summary is presented in section 2.4.

2.2 Deterministic modelling of sea clutter

2.2.1 Introduction

The traditional approach has been to model sea clutter as a stochastic process. This approach is discussed in section 2.3. However, recently some researchers have put forward evidence which suggests that sea clutter is a nonlinear deterministic process [9–21, 27]. More specifically, sea clutter has been characterised as a chaotic process in [11, 13, 14, 17–21]. For the research reported in [9–21] the data was collected using a stationary land-based radar pointing in a fixed direction. The same approach was used in the collection of the data sets analysed for the work in this thesis, see Appendix A. It should be noted that pulse compression and polarisation agility were not used in the collection of the data sets analysed in [9–21], however these techniques were used in the collection of some of the data sets (the wavetank data sets) described in Appendix A.

In sections 2.2.2 to 2.2.10 the concept of a chaotic signal, and associated aspects will be discussed. In section 2.2.11 a discussion will be given on the evidence already presented in the literature which suggests that sea clutter is best modelled as a chaotic process. In section 2.2.12 criticisms of the inferences based upon the evidence presented in section 2.2.11 will be considered.

2.2.2 Classification of chaotic and stochastic processes

Haykin and Li [17] classified chaotic and stochastic processes, as shown in Table 2.1.

Process	**Description**
Deterministic	The signal value can be described precisely for all instants of time.
Stochastic	Ensembles of random waveforms with each ensemble being defined by an underlying probability distribution.
Chaotic	The signal mostly has a very irregular (random-like) waveform, but is generated by a deterministic mechanism.

Table 2.1: *Haykin and Li's classification of chaotic and stochastic processes.*

Succinctly put, a chaotic signal is the observation of a nonlinear dynamical deterministic system with at least one positive Lyapunov exponent [17, 18]. A nonlinear dynamical system with at least one positive Lyapunov exponent means that the system is sensitive to initial conditions.

In a paper by Kubin [28] an alternative approach, to that employed by Haykin and Li, was introduced for the classification and analysis of chaotic signals. This allowed for a chaotic signal to be defined without reference to the underlying system responsible for the chaotic signal's generation, as well as clarifying the relationship between a chaotic signal and a stochastic signal. Essentially Kubin viewed a chaotic signal as a stochastic process, where the random aspect of the chaotic signal is introduced in the selection of the initial conditions. The implication of this is that given infinite precision, the initial conditions of a chaotic process could be accurately specified, and the random nature of the chaotic signal would be lost. However, finite precision is only ever available in the real world.

2.2.3 Is deterministic modelling better than stochastic modelling?

The random-like behaviour of measurements taken from physical systems in nature is considered to be due to the interaction of a large number of degrees of freedom in the actual phys-

ical system [21]. For this reason stochastic processes have been used to model the complex, random-like behaviour of measurements from physical systems. However, deterministic chaos which is highly irregular is possible [9] using only a low number of degrees of freedom. Therefore, a deterministic chaotic system with only a small number of degrees of freedom could be used to model the complex, irregular behaviour of physical processes, such as the scattering of electromagnetic waves from the sea surface (*i.e.* sea clutter). Furthermore, using a deterministic model of clutter may lead to a better understanding of the process which produces the clutter [9, 10]. After all, random process theory is an empirical technique for coping with inadequate information about the physical sources responsible for the generation of the process: the theory says nothing about the causes of randomness [9].

2.2.4 Aspects of chaos theory

Before going on to discuss the existing evidence for and against clutter being chaotic, some aspects of chaos theory must be explained. These aspects include a discussion of dynamical systems and state space in section 2.2.5, an explanation of dynamical reconstruction and Takens' embedding theorem in section 2.2.6, and average mutual information in section 2.2.7. These also include a description of the following chaotic invariants [21]: the correlation dimension is described in section 2.2.8, Lyapunov exponents in section 2.2.9, and the Kaplan-Yorke dimension in section 2.2.10. This discussion is also of great relevance to the material on nonlinear prediction in Chapters 4, 5 and 6.

2.2.5 Nonlinear dynamical systems and state space

Let $\zeta_1(t), \zeta_2(t), \ldots, \zeta_d(t)$ denote the state variables of a nonlinear dynamical system, where continuous time t is the independent variable, and d is the order, or dimension, of the system. For convenience, the state variables can be collected into a length d vector $\boldsymbol{\zeta}(t)$, which will be called the state vector of the system. The dynamics of a large class of nonlinear dynamical systems may then be cast in the form of a set of first-order differential equations, written in terms of the state vector as follows,

$$\frac{d\boldsymbol{\zeta}(t)}{dt} = \Psi(\boldsymbol{\zeta}(t)) \tag{2.1}$$

where the vector function $\Psi(.)$ is nonlinear. The dynamics of the nonlinear dynamical system are said to be described by the state vector $\boldsymbol{\zeta}(t)$ evolving in a d-dimensional space, according to the first-order differential vector equation given in equation (2.1). This d-dimensional space is commonly referred to as the state space, or phase space, of the dynamical system.

2.2.6 Dynamical system reconstruction and Takens' embedding theorem

In order to make physical sense [29] of a chaotic time series, the underlying dynamics of the system responsible for the generation of the time series must be reconstructed. Takens [30] proposed a technique which is now known as Takens' embedding theorem [17], and which provides a mathematical basis for the dynamical reconstruction problem. Effectively, Takens' theorem states that dynamical reconstruction of a system can be achieved using a single dimensional subspace of the actual system. Consider a d-dimensional chaotic dynamical system described by the difference equation,

$$\boldsymbol{\zeta}(t+1) = \Psi(\boldsymbol{\zeta}(t)) \tag{2.2}$$

where $\Psi(.)$ is a nonlinear function. Given an initial state $\boldsymbol{\zeta}(0)$, the system trajectory is a sequence of points $\{\boldsymbol{\zeta}(k) = \Psi^k(\boldsymbol{\zeta}(0)), k = 1, 2, ...\}$. Suppose now, that a sequence of measurements $\{x(t) = f(\boldsymbol{\zeta}(t)), t = 0, 1, ...\}$ is made where $f(.)$ is a smooth[1] observation map of the dynamical system. According to Takens' embedding theorem, the dynamics of the original system can be reconstructed using a D_E-dimensional space constructed from the observed data $\{x(t)\}$, using the following reconstruction vector,

$$\mathbf{x}_{Rv}(t) = [x(t), x(t-\tau), \ldots, x(t-(D_E-1)\tau)]^T \tag{2.3}$$

where D_E is known as the embedding dimension, and τ is known as the embedding delay. The reconstruction described by equation (2.3) is possible provided that,

[1] Smooth, here, means at least twice differentiable and continuous. Note that if a function is differentiable at a point it must also be continuous, whereas the converse is false.

$$D_E \geq 2d + 1 \tag{2.4}$$

where d is the order, or dimension, of the dynamical system. It should be noted that equation (2.4) is a sufficient, but not necessary condition for dynamical reconstruction [29].

Essentially, what this means is that the evolution of points $\mathbf{x}_{Rv}(t) \rightarrow \mathbf{x}_{Rv}(t+1)$ in the reconstruction space follows that of the unknown dynamics $\boldsymbol{\zeta}(t) \rightarrow \boldsymbol{\zeta}(t+1)$ in the original space. However, this is not completely correct: the evolution of points in the reconstruction space will only follow that of the unknown dynamics in the original space, so long as the observation map is a diffeomorphism[2].

Haykin and Li [17] state that Takens' embedding theorem not only produces a reconstructed model with the same Lyapunov exponents (Lyapunov exponents are discussed in section 2.2.9) as those of the original system, but also, that reconstruction is possible using the in-phase, quadrature-phase or amplitude component of a coherent (radar) signal. However, Haykin and Li simply justify this statement by saying that Takens' theorem can be applied to any time series (or any component of a radar signal) as long as it is obtained using a smooth (see description above) observation map of the original dynamical system.

For this reconstruction to be achieved, an embedding delay τ and an embedding dimension D_E must be estimated from the observed data $\{x(t)\}$. The average mutual information, described in section 2.2.7, was used to choose a suitable value for τ, for the work in this thesis. The method chosen to estimate D_E was to use the correlation dimension, which is described in section 2.2.8, to estimate the dimension d of the original dynamical system.

2.2.7 Average mutual information

Takens' embedding theorem actually permits the use of any τ in the reconstruction vector $\mathbf{x}_{Rv}(t)$, as long as the observed time series is infinitely long. Unfortunately, in practice there is access only to finite data sequences. τ should be chosen so that it is large enough for $x(t)$ and

[2]Before explaining what a diffeomorphism is, first consider the term homeomorphism. A homeomorphism [31] is an invertible map with a continuous inverse. A diffeomorphism [31] is a map in which both the map and its inverse are smooth.

$x(t-\tau)$ to be independent enough of each other, so that they serve as (independent) coordinates of the reconstruction space, *but*, not so independent as to have no correlation with each other: if the gap is too large, chaos makes $x(t)$ and $x(t-\tau)$ disconnected, or statistically independent [32].

One method of choosing a suitable τ is to use the autocorrelation function [33] (ACF), $R_{xx}(\tau)$,

$$R_{xx}(\tau) = \frac{1}{K}\sum_{t=0}^{K} x(t)x(t+\tau) \tag{2.5}$$

where the mean of $\{x(t)\}$ is removed before the ACF is estimated. Using this method, τ is chosen [12] as the first zero-crossing point of the ACF.

The average mutual information [32] is another method which can be used to select an appropriate embedding delay. It has been suggested [34] that the optimum embedding delay is at the first minimum of the average mutual information $I(\tau)$. The average mutual information between $x(t)$ and $x(t+\tau)$ is defined as,

$$I(\tau) = \sum_{t=1}^{K} P(x(t), x(t+\tau)) \log_2 \left[\frac{P(x(t), x(t+\tau))}{P(x(t))P(x(t+\tau))}\right] \tag{2.6}$$

where $P(x(t))$ is the probability density for measurement $x(t)$, and $P(x(t), x(t+\tau))$ is the joint probability for measurements $x(t)$ and $x(t+\tau)$. If $I(\tau)$ is zero, $x(t)$ and $x(t+\tau)$ are independent.

2.2.8 Correlation dimension

The correlation dimension [17, 35] D_c can be used to estimate the attractor dimension d of a nonlinear dynamical system. This estimate can then be used to obtain a suitable value of D_E, using equation (2.4). The approach adopted for the estimation of the correlation dimension in this thesis is known as the maximum likelihood method [35]. This method, unlike the conventional method for correlation dimension estimation, takes into account additive noise. A discussion of the conventional method is given before the method of maximum likelihood is

explained.

Conventional method

The conventional method uses the correlation integral [35] $C(l)$ to estimate the correlation dimension. The correlation integral is calculated as follows,

$$C(l) = \frac{1}{N(N-1)} \sum_{i,j,i \neq j}^{N} H(l - \|\mathbf{x}(i) - \mathbf{x}(j)\|) \tag{2.7}$$

where $\|.\|$ is a suitable distance metric, $\mathbf{x}(i) = \{x(i), x(i+1), \ldots, x(i+D_E-1)\}$, $\mathbf{x}(j) = \{x(j), x(j+1), \ldots, x(j+D_E-1)\}$, and $H(.)$ is the Heaviside function,

$$H(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{otherwise} \end{cases}$$

The steps required to obtain an estimate for the correlation dimension, using the correlation integral, are as follows,

1. Choose a starting value, say 1, for D_E in the vectors $\mathbf{x}(i)$ and $\mathbf{x}(j)$.
2. The correlation integral $C(l)$ is worked out for all pairs of N samples of the data over a range of values of l, see equation (2.7).
3. The correlation dimension D_c is worked out using,

$$D_c = \lim_{l \to 0} \frac{\log(C(l))}{\log(l)} \tag{2.8}$$

which only holds for small l. For convenience, using $\log$ without a base will be used to imply $\log_{10}$. The range of values of l over which equation (2.8) is valid, is known as the scaling region [35]. Therefore the following power law,

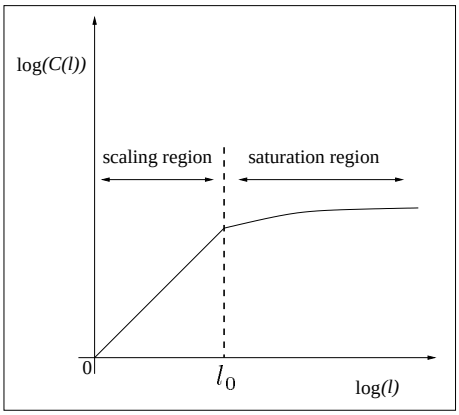


Figure 2.1: *Correlation integral plot: used in the conventional method of correlation dimension estimation.*

$$C(l) \simeq (l)^{D_c} \tag{2.9}$$

is valid for the scaling region. D_c is found by fitting a least squares line to the scaling region of the $\log(C(l))$ vs $\log(l)$ plot, see Figure 2.1. Note that in Figure 2.1, the scaling region is depicted as $0 \leq l \leq l_0$, where l_0 is referred to as the maximum scaling distance. The saturation region is defined as the region where $l > l_0$, and in this region the power law of equation (2.9) does not hold.

4. Repeat steps 2 and 3 for different D_E, and thus collect the D_c value corresponding to each D_E.

5. From the results obtained in step 4, plot D_c vs D_E. If the estimates of D_c converge as D_E increases, then the value to which D_c converges is the estimate of attractor dimension for the system responsible for the generation of $\{x(t)\}$, the observed time series.

Maximum likelihood method

The conventional approach for estimating the correlation dimension from observed data does not take into account the problem of additive noise. The maximum likelihood method of correlation dimension estimation, proposed by Schouten *et al.* [35], does however take into account additive noise.

The maximum likelihood approach accepts the fact that the observed data will be corrupted,

to some extent, by additive noise. In [35] Schouten *et al.* modelled the additive noise process as being bounded in magnitude, with a maximum possible amplitude of $\pm\frac{1}{2}\varepsilon_{max}$. For such a noise process, the observed noise-corrupted data sequence $\{x(n)\}$ is given by,

$$x(n) = \xi(n) + \varepsilon(n)$$

where $\{\xi(n)\}$ is the noise free observed sequence, and $\{\varepsilon(n)\}$ is the noise sequence. The assumption made by Schouten *et al.* [35] is that there is a trajectory that satisfies the true dynamics of the dynamical system, which is sufficiently close to the measured, noise-corrupted trajectory. In principle, $\{\varepsilon(n)\}$ can be any type of noise process [35], but Schouten *et al.* considered the case for noise with correlations much smaller than the time length of the vector used for dynamical reconstruction.

As with the conventional method, a plot of $\log(C(l))$ vs $\log(l)$ is used to estimate the correlation dimension using the maximum likelihood method. For the maximum likelihood method, Schouten *et al.* [35] used the max norm distance measure,

$$\|\mathbf{x}(i) - \mathbf{x}(j)\| = \max_{0\leq k\leq D_E-1} |x(i+k) - x(j+k)| \tag{2.10}$$

in equation (2.7), to find the maximum distance between two vectors, $\mathbf{x}(i)$ and $\mathbf{x}(j)$. The distances that are relevant to the estimation of correlation dimension, using the maximum likelihood method, are given in Table 2.2.

Distance	**Description**
l_ε	maximum noise distance $(= \varepsilon_{max})$.
l_ξ	maximum distance between 2 points on the noise-free reconstructed attractor.
l_x	maximum possible distance between 2 points on the noisy reconstructed attractor.
l_0	maximum scaling distance, *i.e.* the limit of the scaling region.

Table 2.2: *Distances that are relevant to the maximum likelihood method of correlation dimension estimation.*

Schouten *et al.* argue that when a noisy chaotic attractor is embedded in D_E-dimensional space, the noisy part will only be space filling[3] on length scales smaller than the noise scale. When the noise is space filling, *i.e.* for $l \leq l_\varepsilon$, then $C(l)$ scales like l^{D_E}. For length scales bigger than the noise scale, but smaller than the scaling region's upper limit, *i.e.* $l_\varepsilon < l \leq l_0$, then $C(l)$ scales like $(l - l_\varepsilon)^{D_c}$. In other words, the correlation integral can be considered as having two parts,

$$C_1(l \leq l_\varepsilon) \simeq l^{D_E} \tag{2.11}$$

and,

$$C_2(l > l_\varepsilon) \simeq (l - l_\varepsilon)^{D_c} \tag{2.12}$$

Figure 2.2 shows these two regions of the correlation integral for a general, noisy chaotic signal.

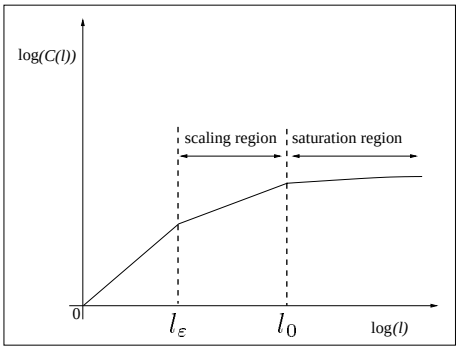


Figure 2.2: *Correlation integral plot: noisy chaotic signal.*

Using the requirements that $C_1(0) = 0$, due to the Heaviside function, $C_1(l_\varepsilon) = C_2(l_\varepsilon) = l_\varepsilon^{D_E}$, for continuity, and that $C_2(l_0) = 1$, see equation (2.7), and, after normalising all distances with respect to l_0, using $r = l/l_0$ and $r_\varepsilon = l_\varepsilon/l_0$, then equations (2.13) and (2.14) are obtained,

[3] In other words a stochastic noise process will tend to fill up all of the reconstruction space, rather than be confined to a bounded region, as is the case for chaotic data.

$$C_1(0 \leq r \leq r_\varepsilon) = r^{D_E} \tag{2.13}$$

and,

$$C_2(r_\varepsilon \leq r \leq 1) = r_\varepsilon^{D_E} + (1 - r_\varepsilon^{D_E}) \left[\frac{(r - r_\varepsilon)}{(1 - r_\varepsilon)}\right]^{D_c} \tag{2.14}$$

When D_E is at least of the order of 50, for moderate values of r_ε (*i.e.* 0 to 0.5), then $r_\varepsilon^{D_E}$ will be very small (less than 10^{-15}), and therefore the right-hand side of equation (2.13) will tend to zero, and equation (2.14) can be re-written as,

$$C_2(r_\varepsilon \leq r \leq 1) = \left[\frac{(r - r_\varepsilon)}{(1 - r_\varepsilon)}\right]^{D_c} \tag{2.15}$$

When l_ε is known *a priori*, all distances can be scaled, and the correlation dimension can be estimated from the slope of $\log(C(l))$ vs $\log(l - l_\varepsilon)$, or from $\log(C(r))$ vs $\log(r - r_\varepsilon)$. In this case, the maximum likelihood correlation dimension D_{ML} can be calculated as follows,

$$D_{ML} = \left[\frac{-1}{M} \sum_{i=1}^{M} \log \left[\frac{r_i - r_\varepsilon}{1 - r_\varepsilon}\right]\right]^{-1} \tag{2.16}$$

where M is the sample size of interpoint distances r_i that satisfy $r_\varepsilon \leq r_i \leq 1$.

In general, l_ε will not be known. In such circumstances, the parameters r_ε and D_c can be estimated from a nonlinear least squares fit to equation (2.15). Once D_c and r_ε have been estimated from equation (2.15), the maximum likelihood correlation dimension estimate D_{ML} is obtained from,

$$D_{ML} = \frac{D_c + r_\varepsilon}{1 - r_\varepsilon} \tag{2.17}$$

Although there is no formal method for deciding on values such as l_0, Schouten *et al.* [35] presented empirical suggestions for these values. A benefit of the maximum likelihood method for estimating correlation dimension, as compared with the conventional method, is that only one embedding dimension (*i.e.* one value of D_E) is required to obtain a correlation dimension estimate. However, the value of D_E used must be sufficiently large. Schouten *et al.* [35] recommend that $D_E \geq 50$.

2.2.9 Lyapunov exponents

The Lyapunov exponents [19, 20] $\{\lambda_i\}_{i=1}^N$ of a nonlinear deterministic dynamical system describe how on average orbits on a system's attractor[4] move apart or together in state space. If the exponents are all zero or negative, the trajectories do not diverge and the system is said to be stable and non-chaotic. If at least one Lyapunov exponent is positive, then the trajectories diverge and the system is said to be unstable and chaotic. This divergence of trajectories implies a sensitivity to initial conditions, which is the hallmark of a chaotic system. Furthermore, the largest positive Lyapunov exponent is inversely proportional to the prediction horizon of a chaotic signal [18].

Wolf *et al.* [37] define the concept of a Lyapunov exponent for a continuous dynamical system in d-dimensional space by considering the long-term evolution of an infinitesimal d-dimensional sphere of initial conditions. They state that the sphere will, in time, evolve into a d-dimensional ellipsoid. The i^{th} one-dimensional Lyapunov exponent λ_i is then defined in terms of the length of the ellipsoid's principal axis $a_i(t)$ as,

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t} \log \frac{a_i(t)}{a_i(0)} \tag{2.18}$$

[4] An attractor is a bounded subset of state space where initial conditions eventually converge on with increasing time. The set of initial conditions that converge to a particular attractor is called the basin of attraction for that attractor [36].

For a d-dimensional system, there are d Lyapunov exponents [21]. The λ_i are conventionally ordered from largest to smallest. A positive Lyapunov exponent obtained from sea clutter data has been used by some researchers as evidence that clutter is chaotic. This evidence is discussed in section 2.2.11. To estimate the largest Lyapunov exponent, the long-term evolution of a single pair of nearby orbits, in state space, is monitored [11, 17, 37].

2.2.10 Kaplan-Yorke dimension

The Kaplan-Yorke dimension is now defined. The reason for the inclusion of this definition is that it was one of the chaotic invariants which were used by Haykin and Puthusserypady [21] to determine if sea clutter is chaotic, see section 2.2.11.

Given a deterministic dynamical system with an attractor dimension of d, and thus d Lyapunov exponents $\{\lambda_1, \lambda_2, ..., \lambda_d\}$, then the Kaplan-Yorke dimension D_{KY} is defined as [19],

$$D_{KY} = K + \frac{\sum_{i=1}^{K} \lambda_i}{|\lambda_{K+1}|} \tag{2.19}$$

where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$, and K is the largest integer for which $\lambda_1 + \lambda_2 + ... + \lambda_K \geq 0$.

2.2.11 The interpretation of sea clutter as a chaotic process

This section discusses the evidence put forward in the literature which suggests that sea clutter is chaotic.

Before discussing the actual evidence of the chaotic nature of sea clutter, it is interesting to note the following observation made by Leung and Lo [12]. Given the exact geometry of the ocean surface, and the angle of incidence of a radar wave, the exact trajectory of the backscattered wave could be determined. However, for a fixed surface geometry, a slight deviation in the angle of incidence may produce a large change in the backscattering process. This sensitivity to initial conditions could be an indication that sea clutter is a chaotic scattering phenomenon.

Now the evidence of chaos in sea clutter is discussed. Given a time series, how can it be determined whether or not it is chaotic? Unfortunately, there is no single criterion for determining whether or not a time series is chaotic [20, 21]. The strategy adopted by Haykin and

Puthusserypady in [21] was to check to see if the following criteria are satisfied:

1. The process should be nonlinear.
2. The attractor (correlation) dimension D_c should converge for increasing embedding dimension (*i.e.* it should have a finite correlation dimension).
3. The dynamics of the system responsible for the generation of the process should be sensitive to initial conditions. This, in turn, means that at least one of the Lyapunov exponents of the process should be positive. The largest Lyapunov exponent determines the horizon within which the time series is predictable.
4. The sum of all the Lyanpunov exponents of the process should be negative for the underlying dynamics to be dissipative (*i.e.* physically realisable).
5. The Kaplan-Yorke dimension D_{KY} should be close in numerical value to the correlation dimension.

The time series analysed should be long enough to obtain reliable chaotic invariant estimates (such as the correlation dimension). However, the time series should also be stationary.

In [21], as well as analysing raw clutter data, Haykin and Puthusserypady preprocessed their clutter data before analysing it for evidence of chaos. The preprocessing included:

- I-Q calibration
- Filtering[5] to reduce measurement and additive noise. A continuity test and differentiability test were carried out to check that the filtering did not affect the underlying dynamics of the clutter data.

The results obtained in [21] for several sea clutter data sets, collected at different geographical locations and during different sea states, are summarised below:

1. The data sets were found to be nonlinear. This was done using three surrogate data tests. Of these three tests, the one which Haykin and Puthusserypady said gives the most

[5] Two different filtering techniques were used: a linear phase lowpass filter and a smoothing filter. The smoothing filter simply averaged every three samples.

convincing proof that a given time series is nonlinear, is one in which the surrogate data sets are filtered white Gaussian noise: the Gaussian white noise sequences are filtered so that they have the same PSD as that of the signal being tested.

2. The D_c value was fractal and it converged to a value between 4.1 and 4.5 for increasing embedding dimension. Furthermore, D_c was found to be independent of radar type[6], radar range, geographic location, pulse duration (*i.e.* resolution), pulse repetition frequency (PRF), like-polarisation (*i.e.* HH[7] or VV[8]) and signal component (*i.e.* I, Q, or amplitude).

3. The Lyapunov spectrum consisted of two positive exponents, one exponent close to zero and 2 or more negative exponents corresponding to an embedding dimension of 5 or more (the number of exponents is equal to the embedding dimension).

   For a given radar type and sea state, the Lyapunov exponents were found to be essentially independent of radar range and signal component.

   For a given radar type the Lyapunov spectrum exhibited significant dependence on sea state. In particular, the magnitude of the positive Lyapunov exponents increased with increasing wave height: which implies less predictability as the sea state increases.

   Haykin and Puthusserypady stated that the clutter is generated by a coupled system of 5 or 6 nonlinear differential equations. They deduced this from the fact that the third Lyapunov exponent λ_3 was consistently 0 for an embedding dimension of 5 or 6.

4. The sum of all Lyapunov exponents was always negative, indicating that the system responsible for the generation of the clutter is dissipative.

5. The computed D_{KY} was always close to the estimated D_c.

The above results presented by Haykin and Puthusserypady in [21] follow on from earlier research [9–18] into the chaotic nature of sea clutter. However, the research in these earlier papers did not deal as thoroughly with the chaotic characterisation of sea clutter as does the work by Haykin and Puthusserypady in [21]. For example, in [11] He and Haykin claimed that the most convincing evidence for the characterisation of a process as chaotic, is the existence of at least

[6] Two different types of radar were used, a coherent instrumentation quality radar, and a commercial noncoherent marine radar.

[7] *i.e.* transmit polarisation horizontal, receive polarisation horizontal.

[8] *i.e.* transmit vertical, receive vertical.

one positive Lyapunov exponent. There are however criticisms of the conclusions reached in [21], these will be discussed in section 2.2.12.

One of the criteria for determining whether clutter is chaotic in [17] was that clutter was found to be locally predictable using a neural network, which was able to reconstruct the underlying dynamics of sea clutter. Indeed such prediction results have been presented in [9, 12–18]. Prediction is discussed further in Chapter 4, and the prediction of sea clutter is discussed in Chapter 5.

Evidence has been presented by Palmer *et al.* in [27], which suggests that sea clutter is the result of a low order dynamical system. However, they do not make the more specific classification that sea clutter is the result of a chaotic system. Palmer *et al.* investigated wind-sea interactions and found evidence to infer that the observed dynamical behaviour of both vertically polarised radar reflections and horizontal surface winds is controlled by a single low-dimensional dynamical system.

2.2.12 Criticisms of the research which claims sea clutter is chaotic

The results in the previous section seem to provide convincing evidence of the chaotic nature of sea clutter. However, it has been pointed out by Davies [38] and Noga [39] that there are some problems with the criteria used by Haykin and Puthusserypady in [21] for determining if sea clutter is chaotic. These problems are now discussed.

Davies [38] points out that Lyapunov exponents are only defined for truly deterministic systems. Therefore, if determinism cannot be demonstrated conclusively, then Lyapunov exponent estimates have little meaning and thus they cannot be used to distinguish between stochastic and chaotic signals.

Furthermore, Davies points out that the surrogate tests used by Haykin and Puthusserypady [21], to test the significance of their results, were generated to have random phase and the same ACF as the original time series. Davies observed that these surrogates represented typical *Gaussian* linear processes. Therefore, the surrogate data tests in [21] rejected the possibility that the data is linear Gaussian. These tests were done on *amplitude* sea clutter data sets, which have only positive values, whereas the surrogate signal fluctuates both above and below zero. To be more specific, quadrature channel clutter returns may have Gaussian statistics, in which case the amplitude returns will be Rayleigh distributed. Amplitude returns cannot be

Gaussian, therefore the rejection of the possibility that Haykin and Puthusserypady's amplitude sea clutter data sets are linear Gaussian is immediately apparent, without testing for it. The stochastic modelling of quadrature channel and amplitude clutter returns is discussed in section 2.3.4.

Davies goes on to suggest other possible surrogate data tests [38]. However, he warns against jumping to the conclusion that data is nonlinear, and points out other possibilities, for example that the data might be linear, with conditional heteroscedasticity[9] of some form.

Noga [39] carried out a simple surrogate data test which highlights the danger of using chaotic invariants such as the correlation dimension and the largest Lyapunov exponent to classify whether or not a time series is chaotic.

Briefly, Noga [39] generated a surrogate data set of a real amplitude sea clutter data set. The surrogate data set was Rayleigh distributed (*i.e.* positive values only), and adapted to match the spectrum of the first 128 samples of the real clutter data. Only the spectrum of the first 128 samples was used for reasons of clutter stationarity.

Noga found that not only did a finite correlation dimension exist for the real sea clutter data ($D_c \simeq 10$), but that a finite correlation dimension was also indicated for the *stochastic* surrogate data ($D_c \simeq 9$). Therefore, no conclusion can be drawn about sea clutter being chaotic from the finite correlation dimension alone. Noga used the conventional method to estimate the correlation dimension. Leung and Lo in [12] did acknowledge that statistical coloured random noise may have a finite correlation dimension, and thus the correlation dimension on its own may not be enough to determine if clutter is chaotic.

Noga [39] also found that he obtained a positive value for the largest Lyapunov exponent of both the real sea clutter data and the surrogate data. Therefore, neither the correlation dimension nor the largest Lyapunov exponent could be used to reach a conclusion on whether or not Noga's sea clutter data is chaotic. This supports Davies' [38] criticisms of using chaotic invariants to categorise a process as chaotic.

Noga used the following formula to estimate the largest Lyapunov exponent λ_1,

[9] A heteroscedastic process is one which has a non-constant prediction error variance.

$$\lambda_1 = \frac{1}{K}\sum_{k=1}^{K} \log \frac{|\mathbf{x}_{Rv}(k+1) - \mathbf{x}_{Rv}(k'+1)|}{|\mathbf{x}_{Rv}(k) - \mathbf{x}_{Rv}(k')|} \tag{2.20}$$

where k' is the index of the point nearest $\mathbf{x}_{Rv}(k)$ in the reconstructed state space, and where the averaging is done for K consecutive state space points.

2.3 Stochastic modelling of sea clutter

2.3.1 Introduction

As alluded to in section 2.1, the debate over whether or not sea clutter is chaotic has not yet been resolved. The work in this thesis does not depend upon whether or not sea clutter is deterministic or stochastic. Instead a pragmatic approach has been adopted, which aims to take advantage of the deterministic nature of clutter, if it is deterministic, and if not, it aims to take advantage of the widely accepted non-Gaussian nature of high resolution sea clutter. The traditional approach to modelling sea clutter, stochastic modelling, is now discussed.

2.3.2 Electromagnetic backscattering from ocean waves

Before describing specific stochastic models for sea clutter it is necessary to motivate the introduction of stochastic modelling.

A number of scattering models have been proposed to explain experimental sea clutter observations. The most successful[10] of these models is the *composite surface model* [7]. This model considers the sea surface to be composed of a number of waves of different wavelengths and amplitudes. Using this model sea clutter is thought to only come from sea waves with a wavelength λ_w,

$$\lambda_w = \frac{\lambda_r}{2\cos(\phi_r)} \tag{2.21}$$

[10] In terms of being able to explain and account for experimental observations.

Category	**Parameters**
Environmental	Wave height
	Wind speed
	Length of time and distance over which the wind has been blowing
	Direction of waves (relative to radar beam)
	Presence or absence of swell waves (*i.e.* non-wind driven waves)
	Presence or absence of wind driven waves
	Contaminants present on the sea surface (*i.e.* oil)
Radar	Frequency
	Polarisation
	Grazing angle
	Pulse width

Table 2.3: *Parameters that affect sea clutter statistics.*

where λ_r is the radar wavelength and ϕ_r is the grazing angle. Equation (2.21) is usually referred to as the Bragg backscattering resonance condition [7]. This composite scattering model was used by Trunk [40] to explain the non-Gaussian nature of sea clutter returns.

However, as Skolnik points out [7], the composite scattering model is not a complete model of radar backscatter from the sea surface. For example, the composite model works better for a radar using vertical rather than horizontal polarisation. The composite model also does not work so well for low grazing angles.

More recent attempts to better understand the physics of the backscattering of microwaves from ocean surfaces are described in [41, 42]. The important point is that the backscattering process is not fully understood, and so there is not a complete physical model for it. Therefore, stochastic processes have been used to model sea clutter, in order to categorise it, and to try and create effective target detection algorithms that can deal with these sea clutter returns.

2.3.3 Factors that affect sea clutter statistics

The statistics of sea clutter are affected by both environmental and radar parameters. These parameters are tabulated in Table 2.3. In addition to those parameters listed in Table 2.3, it should be noted that the proximity to the shore can also affect sea clutter statistics, for example, shallow water currents and reflections from the land can have an impact on the statistics. Table

2.3 is intended only to highlight the number of factors which may influence the statistics of sea clutter. A general discussion on how the parameters in Table 2.3 affect sea clutter returns is given in [7, 43]. For the more specific case where the clutter is assumed to be modelled by the compound K-distribution (see section 2.3.4), Ward *et al.* [5, 6] provide a discussion of the effect that some of the factors in Table 2.3 have on the parameters of the compound K-distribution model.

Other factors that affect the modelling of sea clutter returns are thermal noise [44, 45], and multi-path effects [5, 46]. These two factors are beyond the scope of this research and will not be considered further in this thesis.

2.3.4 Single point statistics modelling

The effect that pulse width τ_r and grazing angle ϕ_r have on clutter statistics is now discussed in conjunction with a presentation of the stochastic models used to represent sea clutter. It has been found that sea clutter collected at low resolutions (*i.e.* long τ_r) and/or at high grazing angles is Gaussian [7, 8]. To give the reader an idea of what a *high* resolution is, and what a *low* grazing angle is, Farina *et al.* [47] define a high resolution as $\tau_r < 0.5\mu s$ and a low grazing angle as $\phi_r < 5^o$.

The in-phase and quadrature-phase clutter returns are then given by the Gaussian probability density function (pdf),

$$P_G(g|\sigma,\mu) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[\frac{-(g-\mu)^2}{2\sigma^2}\right]; \quad -\infty \leq g \leq \infty \tag{2.22}$$

where μ is the mean and σ^2 is the variance. Assuming that the in-phase and quadrature-phase samples (g_I, g_Q) are independent Gaussian random variables with zero mean and equal variances then the corresponding amplitude returns a_R (where $a_R = \sqrt{({g_I}^2 + {g_Q}^2)}$, are Rayleigh distributed [48],

$$P_R(a_R|\sigma^2) = \frac{a_R}{\sigma^2}\exp\left[\frac{-a_R^2}{2\sigma^2}\right]; \quad a_R \geq 0 \tag{2.23}$$

where σ is known as the scale parameter of the Rayleigh distribution, and the mean of a_R is given by $\sigma\sqrt{\frac{\pi}{2}}$. The power or intensity, $I_e = a_R^2$, has an exponential distribution,

$$P_e(I_e|\sigma^2) = \frac{1}{2\sigma^2}\exp\left[\frac{-I_e}{2\sigma^2}\right]; \quad I_e \geq 0 \tag{2.24}$$

If the pulse width of a radar τ_r is decreased then the clutter power collected by the radar is reduced and the signal to clutter ratio (SCR) is improved. However, this benefit may be offset by the change in the statistics of the clutter returns as τ_r is reduced. This is because at high resolutions (*i.e.* short τ_r) and/or low grazing angles, sea clutter returns have been found to be non-Gaussian [8]. The non-Gaussian nature of these clutter returns is significant as they are often characterised by frequently occurring large signal values (or spikes) [49], which may be mistaken for target signals in radar detectors. The correct classification of such non-Gaussian distributions is that they are *heavy-tailed* [50], *i.e.* there is more mass in the tails of these distributions than there is in the distribution of Gaussian data with the same variance, or power, as the non-Gaussian data.

Several models have been used for non-Gaussian amplitude sea clutter returns. Amongst the more well known models are the log-Normal [51] distribution with scale parameter m and shape parameter n,

$$P_{lN}(a_{lN}|m,n) = \frac{1}{a_{lN}n\sqrt{2\pi}}\exp\left[\frac{-\left[\log\left(\frac{a_{lN}}{m}\right)\right]^2}{2n^2}\right]; \quad a_{lN} \geq 0 \tag{2.25}$$

and the Weibull [51] distribution with scale parameter r and shape parameter s,

$$P_W(a_W|r,s) = \frac{sa_W^{s-1}}{r^s}\exp\left[\frac{-{a_W}^s}{r}\right]; \quad a_W \geq 0 \tag{2.26}$$

Both of these models have simply been used to fit to clutter data, and do not have any physical justification for why they are a good fit to sea clutter data. However, the K-distribution,

$$P_K(x|\nu,c) = \frac{4c}{\Gamma(\nu)}(cx)^{\nu} K_{\nu-1}(2cx); \quad x \geq 0 \tag{2.27}$$

where $\Gamma(.)$ is the gamma function, K_ν is a modified Bessel function of order ν, c is the scale parameter, and ν is the shape parameter, has been shown to provide a good fit to sea clutter amplitude returns, and it has been shown (see section 2.3.6) to have some physical justification for its use. The limiting case of the K-distribution, for $\nu \to \infty$, is the Rayleigh distribution [8]. As $\nu \to 0$, the K-distribution tail gets heavier and thus the K-distributed variates have more frequently occurring large values (or spikes). This is illustrated below in Figure 2.3, where the effect of varying the shape parameter of the K-distribution is shown, using normalised pdf plots.

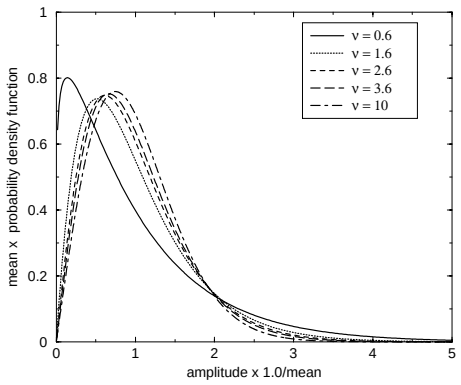


Figure 2.3: *Plots of the normalised K-distribution pdf for the following values of shape parameter: 0.6,1.6,2.6,3.6 and 10. The normalised pdf allows the effect that the shape parameter has on the K-distribution to be shown: as the shape parameter goes to zero, the tail of the K-distribution becomes heavier.*

2.3.5 Evidence that amplitude sea clutter returns are modelled by the log-Normal, Weibull, and K distributions

Trunk and George [52] found that the log-Normal distribution was a fairly good fit to their sea clutter data. However Valenzuela and Laing [53] found that the sea clutter they analysed had a distribution that lay between the Rayleigh and log-Normal distributions.

Schleher [54] introduced the Weibull distribution as an intermediate model between the log-Normal and Rayleigh cases, and he showed that it provided a good fit to some amplitude sea

clutter returns. Fay *et al.* [55] also showed that the Weibull distribution was a good fit to sea clutter returns.

Jakeman and Pusey [56] introduced the K-distribution as a model for non-Rayleigh sea clutter returns. In [57] Jakeman and Pusey showed that the K-distribution was a good fit to amplitude sea clutter returns. More recently Nohara *et al.* [58] showed that the K-distribution was a good model for both like-polarised and cross-polarised sea clutter amplitude returns. They used a coherent dual-polarised X-band radar [59]. They also showed that the HH (*i.e.* transmit horizontal, receive horizontal) data was in general spikier (*i.e.* its distribution had a heavier tail) than the VV data. In [60] Haykin and Nohara built on the work in [58], and they stated that although the K-distribution model was found to be a good fit to like-polarised clutter returns, the K-distribution plus thermal noise model [44, 45] was a better fit to the cross-polarised clutter returns. In [61, 62] Farina *et al.* analysed data collected using the radar described in [59]. They found that the VV amplitude returns were well modelled by the K-distribution, and the cross-polarised amplitude returns fitted a K-distribution plus thermal noise model. However, the HH amplitude returns were found to be a better fit to the log-Normal distribution than to the K-distribution.

Hair *et al.* [49] and Bouvier *et al.* [63] also reported that the K-distribution was a good model for amplitude clutter returns.

Chan [64] found that the best model for amplitude returns was almost always the K-distribution. Chan used a modified chi-square test to reach this conclusion, which only analysed the tail of the amplitude distribution of clutter returns.

To summarise, results have been presented in the literature which have shown that the log-Normal, Weibull and K distributions can all provide reasonable fits to observed sea clutter amplitude returns.

2.3.6 The compound K-distribution

As discussed in the previous section, the Weibull, log-Normal, and K distributions can all provide reasonable fits to observed clutter amplitude returns. However, of perhaps more importance than the choice of amplitude distribution, from the perspective of detection performance assessment [65], is the modelling of the correlation properties of clutter. The compound K-distribution pdf equations and correlation properties are now discussed. Physical justification

for the use of the compound K-distribution to represent sea clutter is also considered.

Ward [22] introduced the compound form of the K-distribution. This model consists of 2 components [66]. The first is a fast fluctuating Rayleigh component with amplitude a_R,

$$P_R(a_R|\sigma^2 = \frac{2a_c^2}{\pi}) = \frac{\pi a_R}{2a_c^2} \exp\left[\frac{-\pi a_R^2}{4a_c^2}\right]; \quad a_R \geq 0 \tag{2.28}$$

that has a slowly varying mean amplitude value a_c, which is the second component of the compound K-distribution, and this is chi-distributed,

$$P_c(a_c|\nu, b) = \frac{2b^{2\nu}}{\Gamma(\nu)} a_c^{2\nu-1} \exp\left[-b^2 a_c^2\right]; \quad a_c \geq 0 \tag{2.29}$$

where $\Gamma(.)$ is the gamma function, b is the scale parameter and ν is the shape parameter. The mean amplitude value a_c is often referred to in terms of power z where,

$$z = a_c^2 \tag{2.30}$$

The pdf of z is gamma distributed and is given by,

$$P_\Gamma(z|\nu, b) = \frac{b^{2\nu}}{\Gamma(\nu)} z^{\nu-1} \exp\left[-b^2 z\right]; \quad z \geq 0 \tag{2.31}$$

The overall K-distributed amplitude distribution can be written as [66]:

$$P_K(x) = \int_0^\infty P_R(a_R|\sigma^2 = \frac{2a_c^2}{\pi}) P_c(a_c) dy \tag{2.32}$$

which can be re-written [66] as equation (2.27), which is repeated below for convenience.

$$P_K(x|\nu, c) = \frac{4c}{\Gamma(\nu)}(cx)^{\nu} K_{\nu-1}(2cx); \quad x \geq 0$$

The scale parameters of the chi distribution and the K-distribution are related as shown in equation (2.33).

$$c = b\sqrt{\frac{\pi}{4}} \tag{2.33}$$

Note that a model such as the compound K-distribution is sometimes referred [62] to as a compound Gaussian distribution, as it has a complex Gaussian (or Rayleigh amplitude) component which is modified in some way by another distribution. In the case of the compound K-distribution, the modification is a chi-distributed mean level.

The compound K-distribution model relates very well to the composite scattering model discussed in section 2.3.2. Using the composite scattering model, and according to equation (2.21), at high resolutions radar backscatter is due to small capillary waves [7]. As Trunk pointed out [40], if the radar returns do come from the small capillary waves, then one can also expect these returns to be affected by larger waves which physically move the smaller capillary waves. This interaction of waves is simplistically illustrated in Figure 2.4.

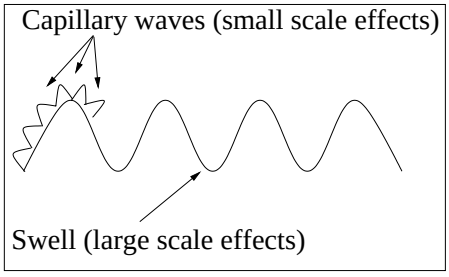


Figure 2.4: *Composite scattering model. Small capillary waves ride on top of, and are affected by, large scale waves such as swell.*

This interaction of large and small scale effects is incorporated by the compound K-distribution model, as follows. The compound K-distribution model represents sea clutter as a Rayleigh process with a varying (chi distributed) mean. Assuming this model is accurate, this would

suggest that in any individual resolution cell there are enough scatterers for the central limit theorem to hold. Given this fact, the non-Rayleigh amplitude returns may then be explained by a bunching of scatterers due to swell (*i.e.* large scale waves), rather than being due to just a small number of effective scatterers [5]. The Rayleigh component of the compound K-distribution model is therefore associated with (*i.e.* it represents scattering from) small scale effects such as spray, foam and capillary waves [67]. The chi component of the compound K-distribution model is associated with large scale effects such as swell and breaking waves [67], which cause the bunching of scatterers described above.

The Rayleigh component of the compound K-distribution model is often referred to as the speckle component. Similarly, the chi component is often referred to as the underlying component. The chi component is also often referred to as the gamma component (*i.e.* the chi component in terms of power).

The compound K-distribution model allows the correlation properties of sea clutter to be easily modelled [22]. The Rayleigh component of the compound model is represented as a fast fluctuating process with a much shorter correlation length than that of the chi component. The Rayleigh component is typically modelled as having a correlation length of a few milliseconds, whereas the chi component will usually have a correlation length of a few seconds [5, 67].

The K-distribution and the compound K-distribution have been accepted as good models for clutter returns based on the analysis of observed data. This empirical work has motivated a theoretical justification of the use of these distributions as good clutter models. Theoretical proposals that are based on the interaction of the electromagnetic waves with the sea surface become less appealing when it is found that the K-distribution is a good model for amplitude land clutter returns [68], and when it is found that the compound K-distribution is a good model for texture in synthetic aperture radar (SAR) images of land [69]. Jakeman [70] proposed that the K-distribution is a result of a bunching of scatterers caused by a birth-death-migration process. Whilst Ward *et al.* [5] showed that the gamma distribution is only an approximation to the power of the underlying component of sea clutter, and that it actually results in a model which slightly under-estimates the moments of the real clutter data.

Following on from the result found by Ward *et al.* [5], that the gamma distribution is not an exact model for the underlying component of sea clutter, Noga [39] found evidence which suggested that the underlying component of sea clutter (assuming that a compound Gaussian

model was still appropriate) was better modelled by the log-Normal distribution than by the gamma distribution.

Some authors [71, 72] have suggested using a more generalised compound model to represent sea clutter. This generalised model uses generalised gamma distributions, see equation (2.34), for both the speckle and the underlying components of sea clutter.

$$P(z|\nu,\alpha,\beta) = \frac{|\beta|}{\alpha\Gamma(\nu)}\left(\frac{z}{\alpha}\right)^{\beta\nu-1}\exp\left[-\left(\frac{z}{\alpha}\right)^{\beta}\right] \tag{2.34}$$

2.3.7 Evidence that the compound K-distribution is a good model for sea clutter returns

Several researchers have analysed sea clutter for evidence that the compound K-distribution is a good model for it.

The Rayleigh or speckle component can be extracted by analysing a block of sea clutter data over which the chi component is effectively constant. Baker *et al.* [73], Ward *et al.* [5], and Farina *et al.* [61, 62] all showed evidence of a Rayleigh component of sea clutter, by analysing blocks of data (in individual range cells) corresponding to time periods of 200ms, 250ms, and 250ms respectively. Hair *et al.* [49] also showed evidence of a Rayleigh component of sea clutter, but they only specified that the time period used to isolate this component was short compared to the fluctuations of the mean level of the clutter.

The chi or underlying component can be extracted by averaging out the Rayleigh component. Baker *et al.* [73], Ward *et al.* [5] and Farina *et al.* [61, 62] all showed evidence that the power of the underlying component was gamma distributed. They did this by averaging out the Rayleigh component over time lengths of 250ms, 250ms, and 32ms respectively.

As already discussed in section 2.3.6, one of the attractions of the compound K-distribution model is the ability it provides to account for the correlation properties of sea clutter. The spatial correlation (range) properties of the underlying component were discussed in [74]. In [61] the correlation properties of the VV amplitude returns in individual range cells (*i.e.* temporal correlations) were investigated by Farina *et al.* As mentioned in section 2.3.5, the VV amplitude returns in [61] were the only returns found to be well modelled by the K-distribution, and so

they were the only ones analysed for evidence of the two components of the compound K-distribution model. Farina *et al.* found that two different components each with a different correlation length were present. The first component had a correlation length on the order of only milliseconds, which was associated with the speckle component of the compound model. The second component had a correlation length on the order of seconds, this was associated with the underlying component of the compound model.

Having presented evidence from the literature which supports the claim that the compound K-distribution is a good model for sea clutter, it seems apt to include the following quote from Ward and Watts [8]:

> "We do not claim that the K-distribution is an exact fit to all measurements of radar sea clutter, but rather the model provides a very good approximation to most conditions and, in particular, allows a realistic treatment of problems associated with correlation properties."

For the reasons outlined in the quote above, the compound K-distribution model has become the popular choice of sea clutter model within the radar community. Therefore, the compound K-distribution model was chosen as the stochastic sea clutter model for this thesis. In Chapter 3, the topic of the simulation of correlated compound K-distributed data will be discussed, for the purpose of generating compound K-distributed surrogate data sets.

Finally, to conclude this section on the compound K-distribution model, it should be pointed out that as technological advances make higher and higher resolution radars possible, there may eventually come a time when there might not be enough scatterers per resolution cell for the central limit theorem to hold in each cell. When this occurs the compound K-distribution model may no longer be a good representation of sea clutter returns. Perhaps by this time a deterministic model of sea clutter will be available, and there will not be the need to seek out a new stochastic model to replace the compound K-distribution.

2.3.8 Other models

Distributions other than the log-Normal, Weibull, and K have been used to model sea clutter returns. These other distributions include the contaminated-Normal distribution [52] and the log-Weibull distribution [75].

Recently, the alpha-stable distribution has been used to try to model sea clutter. In [76] Pierce found that the positive alpha-stable distribution closely modelled sea clutter data. However, in [77], Ilow and Leung found that the K-distribution was a better fit to amplitude clutter returns than the envelope of a circularly symmetric alpha-stable distribution. This comparison was made in the tail region of the pdf.

It has been suggested [78–81] that sea clutter could be modelled as a spherically invariant random process (SIRP). This is discussed in more detail in Chapter 3. In [39, 82] conditional heteroscedastic processes were put forward as models for sea clutter. Finally, Azzarelli [83] proposed a class of non-Gaussian models for sea clutter which is based on the concept of a fluctuating number of Gaussian scatterers.

2.4 Summary

In this chapter a discussion was presented on nonlinear deterministic systems and associated aspects of chaos theory. The work done by researchers during the last decade which has provided some evidence to suggest that sea clutter is better modelled as a chaotic rather than as a stochastic process, was reviewed. Criticisms of this research were presented. It was pointed out that the debate over which model is more appropriate, a chaotic or a stochastic model, has not yet been resolved. For the design of radar target detectors, the prevailing method within the radar community is to use a stochastic model to account for sea clutter. A discussion of the stochastic models used for both low and high resolution radar returns from the sea surface was presented. In particular, attention was drawn to the compound K-distribution, which is the most popular stochastic model for sea clutter data. This model has not only been shown to provide a reasonable pdf fit to sea clutter returns and to have some physical justification for its use, but it is also able to take into consideration the correlation properties of sea clutter.

Chapter 3
The simulation of correlated compound K-distributed data

3.1 Introduction

In general it is difficult to evaluate the performance of the optimal radar target detector[1] analytically when the clutter samples are both correlated and have a non-Gaussian pdf [84]. This is why it is necessary to be able to model and simulate sea clutter data. The simulated sea clutter allows the performance of target detection algorithms to be quantified against controlled clutter backgrounds [85, 86].

For the work in this thesis, surrogate compound K-distributed data was required, to investigate whether the compound K-distribution is an adequate model for the sea clutter analysed. This was thought to be relevant, particularly in view of the recent claim made that sea clutter is better described by a chaotic process than a stochastic process, as was discussed in Chapter 2.

This chapter is structured as follows. In section 3.2 the multiplicative nature of the K-distribution is discussed. In section 3.3 an overview of existing strategies to simulate correlated K-distributed clutter is presented. The selection of a suitable simulation technique is discussed in section 3.4. In section 3.5 a simulation technique to generate correlated K-distributed surrogate data, with the same compound K-distributed components as a real clutter data set is presented. In section 3.6 a method for obtaining estimates of the compound K-distributed components of real sea clutter data is described. A chapter summary is presented in section 3.7.

3.2 The multiplicative nature of the K-distribution

It was explained in section 2.3.6 that the compound K-distribution is made up of a Rayleigh $a_R(n)$ component with a chi-distributed $a_c(n)$ mean, *i.e.* $P_R(a_R|\sigma^2 = \frac{2a_c^2}{\pi})$, see equation

[1] Radar target detectors and their performance evaluation are discussed in Chapter 7.

(2.28). In section 2.3.4 it was described how a Rayleigh process could be obtained from a complex Gaussian sequence. Using this knowledge it is clear that a complex K-distributed variate $(x_I(n) + jx_Q(n))$ is equivalent to a complex Gaussian variate $(g_I(n) + jg_Q(n))$ where, $P_G(g_I|\sigma_I = \sqrt{\frac{2}{\pi}}a_c, \mu = 0)$, and $P_G(g_Q|\sigma_Q = \sqrt{\frac{2}{\pi}}a_c, \mu = 0)$.

Software routines exist [87] for the generation of white, zero mean, unit variance Gaussian processes. From such processes, complex K-distributed variates may be generated as follows,

$$x_I(n) + jx_Q(n) = (g'_I(n) + jg'_Q)\sqrt{\frac{2}{\pi}}a_c(n) \tag{3.1}$$

where $g'_I(n)$ and $g'_Q(n)$ are white, zero mean, unit variance Gaussian variates, and $a_c(n)$ is a chi-distributed variate from $P_c(a_c|\nu, b)$, see equation (2.29). Amplitude K-distributed data $x(n)$ is obtained very simply as,

$$x(n) = \sqrt{x_I^2(n) + x_Q^2(n)} \tag{3.2}$$

The multiplicative nature of equation (3.1) underpins the simulation schemes discussed in this chapter: a complex Gaussian sequence multiplied by a chi-distributed sequence is used to produce a complex, compound K-distributed sequence.

As mentioned in section 2.3.6, the complex Gaussian (or Rayleigh amplitude) speckle component of the compound K-distribution model is associated with correlations on the order of a few milliseconds, whilst the chi-distributed component is usually correlated for periods of a few seconds. Such correlations can be accounted for in the multiplicative simulation technique, described by equation (3.1), by correlating either or both of the compound components of K-distributed sea clutter. Filtering of Gaussian processes to achieve a desired power spectral density (PSD) is well understood. However, obtaining non-Gaussian sequences with both the desired pdf and PSD is a difficult problem [88]. The difficulty, therefore, in obtaining the desired complex, correlated compound K-distributed variates is in obtaining a chi-distributed sequence with both the desired pdf and PSD. A number of approaches to produce such sequences are considered in section 3.3.

3.3 Overview of existing simulation techniques

3.3.1 Introduction

This section presents an overview of the different options available to simulate compound K-distributed data. As mentioned in section 3.2, the central difficulty in the simulation of such data is in the generation of chi-distributed data with a specified PSD. In section 3.3.2 a simulation technique based on spherically invariant random processes is described. However, it is highlighted that this technique only takes into consideration the correlation properties of the speckle (*i.e.* Gaussian) component of sea clutter. In sections 3.3.3 to 3.3.8, techniques are described which specifically focus on the generation of gamma distributed data with a specified PSD. As was discussed in section 2.4.6, the chi-distributed component of the compound K-distribution is often referred to in terms of gamma power, in the literature. A chi-distributed variate is obtained from a gamma variate, by taking the square root of the gamma variate. The advantages and disadvantages of each correlated gamma simulation technique are described.

3.3.2 Spherically invariant random process (SIRP)

As was discussed in section 2.3.8, it has been suggested that sea clutter can be modelled as a spherically invariant random process [79, 81, 84, 89] (SIRP). More specifically, Conte and Longo [79] have suggested that the temporal returns within one range cell can be modelled as a SIRP.

The simulation of sea clutter based on the SIRP model is achieved as follows. If $\mathbf{x}_N(n) = [x(n+1), x(n+2), ..., x(n+N)]^T$ denotes a spherically invariant random vector (SIRV), $\mathbf{g}_N(n) = [g(n+1), g(n+2), ..., g(n+N)]^T$ is a Gaussian random vector with zero mean and covariance matrix $\mathbf{M}$, and s is a non-negative random variate with pdf $P(s)$, then the sea clutter time series $\{x_N(n)\}$ in one range cell can be simulated as a sequence of SIRV's,

$$\{x(n)\} = \{\mathbf{x}_N^T(0), \mathbf{x}_N^T(N), \mathbf{x}_N^T(2N), ...\} \tag{3.3}$$

where $\mathbf{x}_N^T(n) = \mathbf{g}_N^T(n)s$.

The vector $\mathbf{g}_N^T(n)$ and the variate s are assumed to be statistically independent. The pdf of

$\mathbf{x}_N(n)$ conditioned on s is given by [84],

$$P(\mathbf{x}_N(n)|s) = (2\pi)^{\frac{-N}{2}}|\mathbf{M}|^{\frac{-1}{2}}s^{-N}\exp\left[\frac{-H}{2s^2}\right] \tag{3.4}$$

where H is a non-negative quadratic form given by $H = \mathbf{x}_N^T(n)\mathbf{M}^{-1}\mathbf{x}_N(n)$ and $|\mathbf{M}|$ denotes the determinant of the covariance matrix $\mathbf{M}$. The pdf of $\mathbf{x}_N(n)$ is given by

$$P(\mathbf{x}_N(n)) = (2\pi)^{\frac{N}{2}}|\mathbf{M}|^{\frac{-1}{2}}h_N(p) \tag{3.5}$$

where

$$h_N(p) = \int_0^\infty s^{-N}\exp\left[\frac{-p}{2s^2}\right]P(s)ds \tag{3.6}$$

The pdf $P(s)$ of the random variable s is called the characteristic pdf of the SIRV $\mathbf{x}_N(n)$. Different characteristic pdf's can be used to yield different non-Gaussian pdf's for the clutter sequence, equation (3.3). Details of the required characteristic function to generate K-distributed samples, as well as the characteristic functions required to generate many other different types of non-Gaussian samples, are given in [81, 84]. The closure property of SIRV's allows correlation to be easily introduced into the Gaussian vector $\mathbf{g}_N(n)$ [84]. Additionally, the SIRP model allows independent control of the pdf and correlation function.

Effectively, this SIRP representation of sea clutter models the correlation properties of the Gaussian (speckle) component, but those of the underlying component are not modelled.

3.3.3 Blacknell's simulation technique

Blacknell [85] discussed a technique which uses moving average filters to produce correlated gamma variates.

Blacknell's approach was to correlate an uncorrelated gamma process using a moving average

filter (MAF) in order to produce a correlated gamma process. Independent gamma processes were used with different MAF's to produce a number of independent correlated gamma sequences. These correlated gamma sequences were then added to produce the final correlated gamma sequence which had the desired pdf and PSD. This method used moment generating functions to set the correlation properties, the mean, and the order parameter of the final correlated gamma sequence.

Blacknell showed simulation results for the case where the first three correlation lags of the final correlated gamma sequence were specified, and demonstrated that the technique produced data which had the desired correlation properties, as well as the desired sample moments (a discussion of sample moments, and their estimation is given in Appendix B). In theory, a gamma process with any shape and scale parameters, can be generated using this technique. However, the main limitation with this technique is the increase in complexity as the correlation length (*i.e.* the number of lag terms) to be specified, of the final gamma sequence, is increased. Blacknell states that the source of the complexity is a

> "...combinatorial explosion in the number of possible moving average filters as the size of the largest non-zero [correlation] lag increases."

A further limitation of this technique is that anti-correlations (*i.e.* negative correlations) are not possible.

3.3.4 Oliver and Tough's simulation techniques

Oliver and Tough [90] proposed two techniques to generate correlated gamma variates. Their first method is based on the integration of a stochastic differential equation. This technique is able to produce gamma samples with the correct moments, for any shape and scale parameters, but it is restricted to an exponential correlation function.

The other approach is similar to the method reported by Blacknell [85] (see section 3.3.3): a linear filter is used to introduce the desired correlation properties to an uncorrelated gamma process. However, unlike Blacknell's method, the data produced using this technique is only approximately gamma distributed, since only the second-order correlation moments are represented precisely.

3.3.5 Armstrong and Griffiths' simulation technique

Armstrong and Griffiths [91] described a technique for generating correlated gamma deviates from correlated Gaussian deviates.

A gamma process, equation (2.31), can be obtained from Gaussian processes, equation (2.22), in the following way. Define the random variable $z(n)$ as,

$$z(n) = \sum_{j=1}^{2\nu} g_j^2(n) \tag{3.7}$$

where the $g_j(n), j = 1, 2, ...2\nu$, are statistically independent and identically distributed Gaussian random variables with zero mean and variance σ^2. Then $z(n)$ is a random variable from a chi-square process, equation (3.8), with 2ν degrees of freedom [92],

$$P_\chi(z|2\nu, \sigma^2) = \frac{1}{\sigma^{2\nu} 2^\nu \Gamma(\nu)} z^{\nu-1} \exp\left[\frac{-z}{2\sigma^2}\right]; \quad z \geq 0 \tag{3.8}$$

A gamma process can be obtained from a chi-square process due to the fact that a gamma process of order ν (which is equivalent to saying it has a shape parameter equal to ν) is equivalent to a chi-square process with 2ν degrees of freedom. This can be seen by comparing equation (2.31) with equation (3.8). As can be seen from these two pdf's, the variance σ^2 of the Gaussian processes can be used to control the scale parameter b of the resultant gamma process,

$$b = \left(\frac{1}{2\sigma^2}\right) \tag{3.9}$$

The main limitation with this method is that only half-integer or integer order gamma processes may be generated, as the order of the gamma process is ν, where ν is half the necessarily integer number of Gaussian variates used to produce each gamma variate. To overcome this difficulty Armstrong and Griffiths suggested using a nonlinear transformation to convert a gamma process of restricted order, to a gamma process of any order.

Armstrong and Griffiths also derived an expression, involving an infinite sum, for the correlation coefficient $E[(z_1, z_2)]$ of the correlated gamma sequence, in terms of the correlation coefficient of the constituent Gaussian processes,

$$E[(z_1 z_2)] = \frac{4(1-\rho^2)^{\nu+2}}{\Gamma(\nu)} \times \sum_{i=0}^{\infty} \rho^{2i} \left[\frac{\Gamma^2(\nu+i+1)}{\Gamma(i+1)\Gamma(\nu+i)} \right] \tag{3.10}$$

where,

$$z_1(n) = \sum_{j=1}^{2\nu} g_j^2(n) \tag{3.11}$$

and,

$$z_2(n) = \sum_{j=1}^{2\nu} g_j^2(n-1) \tag{3.12}$$

and ν is the shape parameter of the gamma process, and the correlation coefficient of the constituent Gaussian processes is given by $0 < \rho < 1$. It should be noted that Armstrong and Griffiths considered the specific case of a decaying exponential ACF for the constituent Gaussian processes.

3.3.6 Marier's simulation technique

Marier [93] also adopted the same sum-of-squares approach as Armstrong and Griffiths [91], to transform Gaussian variates into gamma variates. However, Marier provided a more general analysis than Armstrong and Griffiths, by allowing for any ACF for the constituent Gaussian processes, rather than limiting his analysis to a decaying exponential ACF.

3.3.7 Tough and Ward's simulation technique

Tough and Ward [94] presented a method for generating correlated gamma variates from correlated Gaussian variates. Tough and Ward used the following memory-less nonlinear transformation (MNLT) to go from a zero mean, unit variance Gaussian variate $g'(n)$ to a gamma variate $z(n)$ with shape parameter ν, and scale parameter b,

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{g'(n)} \exp\left[\frac{-g^2}{2}\right] dg = \frac{1}{\Gamma(\nu)} \int_{0}^{z(n)} b^{2\nu} z^{\nu-1} \exp[-b^2 z] dz \tag{3.13}$$

Instead of trying to specify an analytical relationship between the ACF's of the Gaussian and gamma processes through the MNLT, Tough and Ward derived an empirical relationship. They did this by considering a decaying exponential ACF for the Gaussian process. This resulted in the gamma process also having, and being limited to, a decaying exponential function. This technique allows gamma distributed variates to be generated which can have any shape and scale parameters, but which are limited to having a decaying exponential ACF.

3.4 Selection of a simulation technique

The main requirement for a simulation technique for the work in this thesis was that, first and foremost, a gamma (or chi) component could be generated with any shape parameter and any scale parameter. Additionally, it was required that the gamma component could be correlated. It was decided that the accurate specification of the gamma component's ACF using a complicated approach would be compromised in favour of a simple technique which could approximately set the correlation properties of the gamma component. From the simulation techniques listed in sections 3.3.2 to sections 3.3.7, this limited the choice of simulation technique down to Oliver and Tough's stochastic differential method, or Tough and Ward's MNLT method. Tough and Ward's method was chosen in preference to Oliver and Tough's method, because of the scope for the modification of this technique to produce gamma variates with both any pdf, and *any* desired ACF. Indeed, such modifications have recently been proposed [95].

3.5 The simulation of correlated compound K-distributed variates

3.5.1 Introduction

As was explained in section 3.2, complex, correlated compound K-distributed data can be produced by multiplying together a complex Gaussian sequence and a chi-distributed sequence. In section 3.5.2 an overall methodology for generating K-distributed data, with both a correlated speckle component, and a correlated chi-distributed component is presented. As mentioned in section 3.4, Tough and Ward's method was chosen to produce correlated chi-distributed data, and this is incorporated into the overall methodology of section 3.5.2. In section 3.5.3 the simulation of the complex, correlated Gaussian speckle component is considered in closer detail. In section 3.5.4 the simulation of the correlated chi-distributed component is dealt with in some detail. In sections 3.5.5 and 3.5.6, correlated K-distributed data simulation results are presented for two different cases of K-distribution shape parameter.

3.5.2 Overall simulation technique

The overall method to simulate correlated compound K-distributed data is depicted in Figure 3.1.

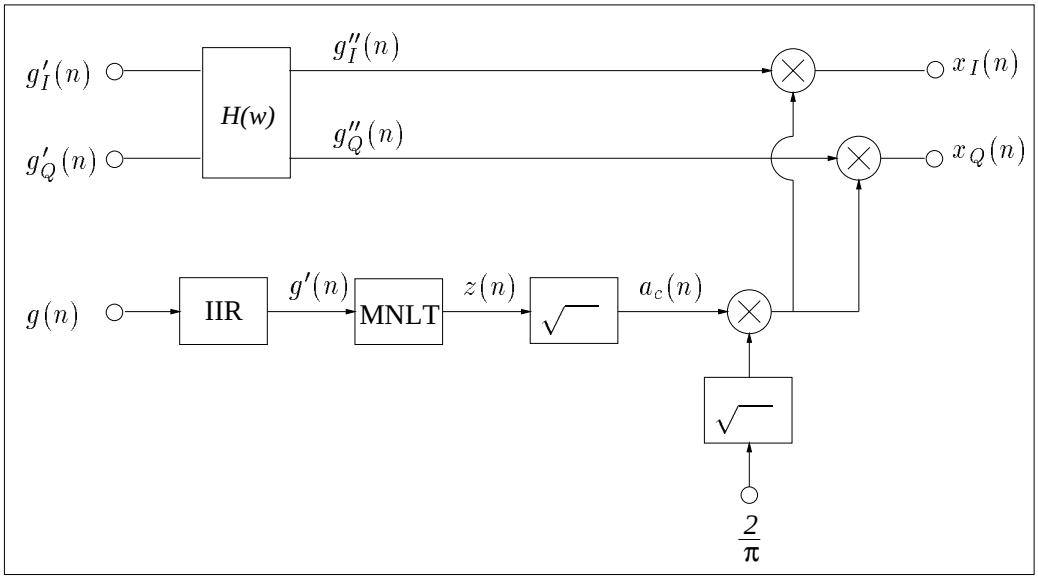


Figure 3.1: *The simulation of correlated K-distributed data.*

A correlated Gaussian speckle process $\{(g''_I(n) + jg''_Q(n))\}$ is obtained by filtering two inde-

pendent, white, zero mean, unit variance Gaussian processes, $\{g'_I(n)\}$ and $\{g'_Q(n)\}$, using the filter $H(w)$. The correlated Gaussian speckle is multiplied by $\{a_c(n)\}\sqrt{\frac{2}{\pi}}$, where $\{a_c(n)\}$ is a correlated chi-distributed sequence, to produce correlated compound K-distributed data $\{(x_I(n)+jx_Q(n))\}$. The correlated chi-distributed sequence is obtained by processing a white, zero mean, unit variance Gaussian process $\{g(n)\}$ through an IIR filter to obtain a correlated, zero mean, unit variance Gaussian sequence $\{g'(n)\}$ which is then passed through the MNLT of equation (3.13), to obtain correlated gamma distributed data $\{z(n)\}$. The correlated chi-distributed sequence $\{a_c(n)\}$ is obtained by taking the square root of the correlated gamma variates.

Tough and Ward's [94] overall methodology for obtaining correlated K-distributed (amplitude) data was to multiply an uncorrelated Rayleigh (speckle) process with chi-distributed data. The reason for this is that these researchers often use frequency agile radars, which remove the speckle correlations from radar data, but which retain the chi component's correlation properties. Therefore, these researchers are not interested in the correlation properties of the speckle component. However, for the work in this thesis, frequency agility was not used, and therefore speckle correlations had to be modelled.

3.5.3 The correlated Gaussian speckle component

In order to obtain a compound K-distributed surrogate data set with the same speckle correlation properties as a real sea clutter data set, the in-phase component $\{x_{realI}(n)\}$ and the quadrature-phase component $\{x_{realQ}(n)\}$ of the real sea clutter data set can be used to estimate the complex ACF of the Gaussian speckle component of the real sea clutter data set, which is the desired complex ACF of the speckle component for the surrogate data set. Therefore, the desired complex ACF $R_{g''g''}$ of the Gaussian speckle component $\{g''_I(n) + jg''_Q(n)\}$ of the surrogate clutter data set can be estimated as,

$$R_{g''g''}(m) = \frac{1}{K}\sum_{n=0}^{K}((x_{realI}(n) + jx_{realQ}(n))(x_{realI}(n+m) - x_{realQ}(n+m))) \quad (3.14)$$

where the means of $\{x_{realI}(n)\}$ and $\{x_{realQ}(n)\}$ are removed before $R_{g''g''}$ is estimated.

Once $R_{g''g''}$ has been estimated, then the complex weights of the linear complex correlation

filter $H(w)$, of Figure 3.1, can be calculated as follows,

- Take the complex fast Fourier transform (FFT) of the estimated complex ACF of the speckle component. This gives the PSD of the speckle component.
- Take the square root of the PSD, to obtain the amplitude frequency response of the speckle component.
- Take the complex inverse FFT of the amplitude frequency response, to obtain the required complex weights for $H(w)$.

3.5.4 The correlated chi-distributed component

The correlated chi-distributed component of the simulated K-distributed data is obtained by correlating a white Gaussian process with an IIR filter, and then passing the output of the IIR filter through a MNLT, and then taking the square root of the MNLT's output. This was depicted in Figure 3.1. More details of the IIR filter and the MNLT are now given.

The IIR filter [96] used to correlate the zero mean, unit variance, Gaussian process $\{g(n)\}$, to produce the correlated, zero mean, unit variance, Gaussian sequence $\{g'(n)\}$, is illustrated in Figure 3.2, and its difference equation is given in equation (3.15).

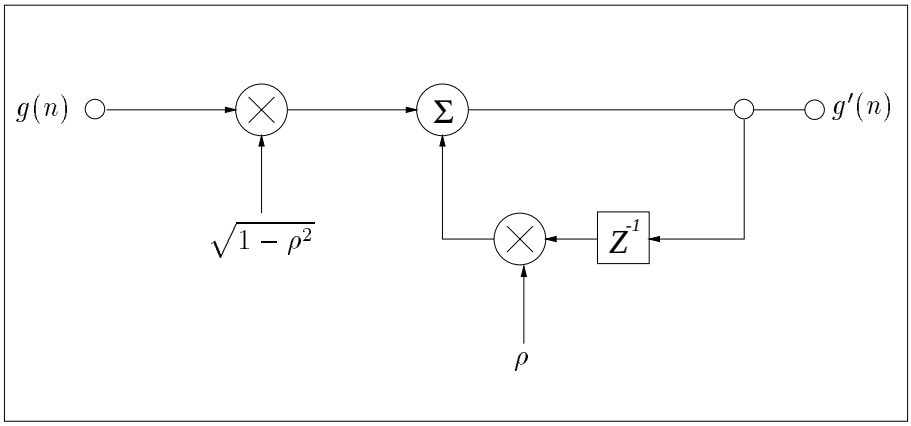


Figure 3.2: *IIR filter used to correlate the zero mean, unit variance, Gaussian process* $\{g(n)\}$ *to produce the correlated Gaussian sequence* $\{g'(n)\}$.

$$g'(n) = \rho g'(n-1) + \sqrt{(1-\rho^2)}g(n) \tag{3.15}$$

The ACF of the correlated Gaussian process $\{g'(n)\}$ from the output of the IIR filter is given in equation (3.16), (see Appendix C for a derivation of this equation),

$$R_{g'g'}(m) = E[g'(n)g'(n+m)] = \rho^m \tag{3.16}$$

where ρ is termed the correlation coefficient of the IIR filter. To attempt to remove the IIR filter transient T_{trans}, the following rule of thumb [48] can be applied,

$$T_{trans} = 5 \times T_c \tag{3.17}$$

where T_c is the time constant, and is defined as,

$$T_c = \frac{1}{\ln(\rho)} \tag{3.18}$$

Tough and Ward used the MNLT, equation (3.13), discussed in section 3.3.7 to transform the correlated Gaussian sequence $\{g'(n)\}$ into a correlated gamma sequence $\{z(n)\}$. They found empirically, that the ACF of the correlated gamma sequence $\{z(n)\}$ is given by,

$$R_{zz}(m) = E[z(n)z(n+m)] = \beta^m \tag{3.19}$$

where

$$\beta = \rho^{\frac{0.15+\nu^{0.7}}{\nu^{0.7}}} \tag{3.20}$$

and ν is the shape parameter of the gamma data. Therefore, using the IIR filter, described above, limits the ACF of the correlated gamma process $\{z(n)\}$ to a decaying exponential. However, some control of the gamma sequence's ACF is afforded, through the correlation coefficient of the IIR filter, ρ.

Tough and Ward [94] recommended using the following MNLT,

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{g'(n)} \exp\left[\frac{-g^2}{2}\right] dg = \frac{1}{\Gamma(\nu)} \int_0^{z(n)} z^{\nu-1} \exp[-z] dz \tag{3.21}$$

instead of the MNLT of equation (3.13). The difference between equations (3.21) and (3.13), is that in equation (3.21) the scale parameter b is set equal to 1. Therefore, if equation (3.21) is used, then the scale parameter of the gamma sequence produced using this MNLT is 1. However, a gamma sequence can be multiplied by $\frac{1}{b^2}$, to result in the gamma sequence having any scale parameter b (see Appendix C). It has been found that using the MNLT of equation (3.21) is computationally much more efficient than using equation (3.13), and therefore the use of equation (3.21), followed by suitable scaling to achieve a desired scale parameter, was the preferred method of generating the correlated gamma data.

Interpolation can be used to solve equation (3.21). In order to use interpolation to solve this equation, a discrete MNLT map between the Gaussian variates on the left-hand side, and the gamma variates on the right-hand side, is required. This is done by evaluating the integral on the right-hand side, for a range of values of $z(n)$.

The software tool *Maple 5 release 3* [97] was used to solve,

$$g'(n) = \frac{1}{\Gamma(\nu)} \int_0^{z(n)} z^{\nu-1} \exp[-z] dz \tag{3.22}$$

for $g'(n)$, given $z(n)$ and ν, in order to produce the required discrete MNLT map. This must be

done for a reasonable range of values of $z(n)$. The selection of a reasonable range of values is considered for 2 different simulation cases in sections 3.5.5 and 3.5.6. The script used to solve equation (3.22) is given in Appendix C.

Given the discrete MNLT map discussed above, a sequence of correlated gamma deviates $\{z(n)\}$ can be simulated using a sequence of correlated Gaussian deviates $\{g'(n)\}$ as follows. Given a correlated Gaussian deviate $g'(n)$ with zero mean and unit variance, the integral on the left-hand side of equation (3.21) is evaluated. This Gaussian integral can be evaluated [87] using the error function $\mathrm{erf}[.]$ [87, 96] as,

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{g'(n)} \exp\left[\frac{-g^2}{2}\right] dg = \frac{1}{2}\left(\mathrm{erf}\left[\frac{g}{\sqrt{2}}\right] + 1\right) \tag{3.23}$$

where,

$$\mathrm{erf}\left[\frac{g}{\sqrt{2}}\right] = \frac{2}{\sqrt{2\pi}} \int_{0}^{g} \exp\left[\frac{-\kappa^2}{2}\right] d\kappa \tag{3.24}$$

Once the left-hand side of equation (3.21) has been solved, interpolation is used to obtain the corresponding gamma variate $z(n)$ for the Gaussian variate $g'(n)$, using the discrete MNLT map discussed above.

In [94] Tough and Ward used cubic B-spline interpolation. However, it was found that a simple 3rd order polynomial interpolation [87, 98] scheme could perform satisfactorily[2], and this was the chosen implementation for the work in this thesis. The interpolation approach adopted included using a discrete MNLT map with many more points than the order of the polynomial: the interpolation was carried out on small (local) sections of the MNLT map [87].

[2] In terms of generating surrogate data with the same moments as the original data.

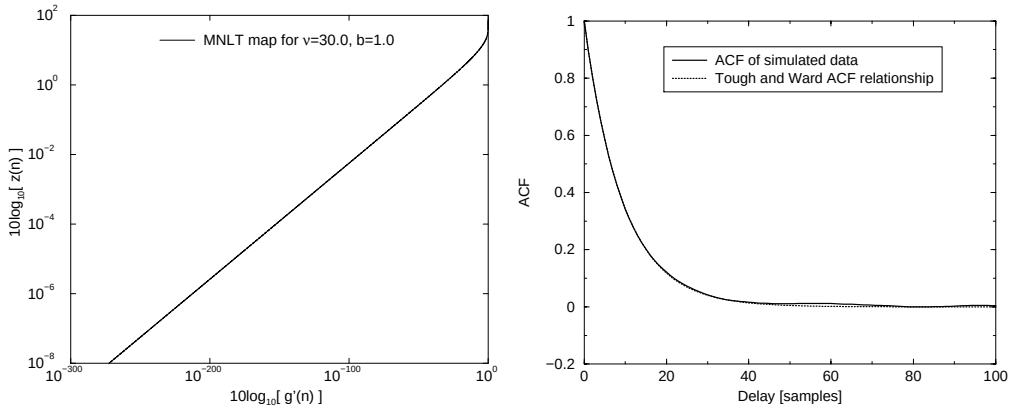
3.5.5 Simulation of correlated K-distributed data with a shape parameter ν equal to 30

This section describes the generation of the MNLT map required to simulate correlated gamma variates with a shape parameter ν equal to 30, and a scale parameter b equal to 1. The correlation coefficient of the IIR filter used to correlate the Gaussian sequence $\{g'(n)\}$, from which the correlated gamma sequence was obtained, was 0.9. The correlated gamma samples were used to obtain correlated complex K-distributed variates, as depicted in Figure 3.1, using an uncorrelated complex Gaussian speckle component. From the complex correlated K-distributed variates, amplitude K-distributed data was obtained using equation (3.2). The K-distributed data had a shape parameter ν equal to 30 and a scale parameter c equal to 0.6267. Note that, despite using an uncorrelated Gaussian speckle component, the K-distributed data generated is still referred to as correlated, since the chi-distributed component was correlated. An uncorrelated Gaussian speckle component, as opposed to a correlated one, was used in this simulation as the interest was, primarily, in demonstrating Tough and Ward's MNLT method for generating correlated gamma variates.

The parameters used in the Maple script, for the generation of the discrete MNLT map, were as follows: *min*=1.0e-8, *max*=76.6, *step*=1.005. This resulted in a MNLT map with 4564 values.

Figure 3.3 shows the MNLT map, and the ACF for the correlated gamma data. Figure 3.3(b) demonstrates the good fit obtained between the ACF of the correlated gamma data, and the empirical ACF derived by Tough and Ward, see equations (3.19) and (3.20).

Sample moments analysis for the correlated gamma and amplitude K-distributed data sets was carried out as follows. One million correlated gamma and amplitude K-distributed variates were generated. Then the first 10 sample moments were estimated for each data set using 100 successive blocks of data, each of length 10,000 samples. The mean and standard deviation were calculated for each sample moment estimate. The moments analysis for the gamma data is given in Table 3.1. The moments analysis for the amplitude K-distributed data is given in Table 3.2. A good agreement was obtained between the desired theoretical moment values and the estimated values obtained from the simulated data.



(a) **(b)**

Figure 3.3: *Generation of correlated gamma data with ν=30, b=1 and IIR filter parameter ρ=0.9, (a) MNLT map, (b) ACF of the correlated gamma data and Tough and Ward's empirical formula for ACF.*

Moment	Theoretical value	Estimated mean	Estimated standard deviation
1	3.00e+1	3.00e+1	0.20e+0
2	9.30e+2	9.29e+2	1.25e+1
3	2.98e+4	2.97e+4	6.07e+2
4	9.82e+5	9.78e+5	2.74e+4
5	3.34e+7	3.32e+7	1.21e+6
6	1.17e+9	1.16e+9	5.32e+7
7	4.21e+10	4.16e+10	2.38e+09
8	1.56e+12	1.54e+12	1.08e+11
9	5.92e+13	5.82e+13	5.06e+12
10	2.31e+15	2.26e+15	2.44e+14

Table 3.1: *Moments analysis of correlated gamma data with ν=30, b=1, and IIR filter correlation coefficient ρ=0.9.*

Moment	Theoretical value	Estimated mean	Estimated standard deviation
1	5.45e+0	5.45e+0	2.60e-2
2	3.82e+1	3.81e+1	3.51e-1
3	3.18e+2	3.17e+2	4.72e+0
4	3.02e+3	3.00e+3	6.80e+1
5	3.19e+4	3.17e+4	1.04e+3
6	3.69e+5	3.66e+5	1.70e+4
7	4.61e+06	4.56e+06	2.93e+05
8	6.19e+07	6.10e+07	5.34e+06
9	8.86e+08	8.66e+08	1.03e+08
10	1.34e+10	1.30e+10	2.06e+09

Table 3.2: *Moments analysis of correlated amplitude K-distributed data with ν=30, c=0.6267, and IIR filter correlation coefficient ρ=0.9.*

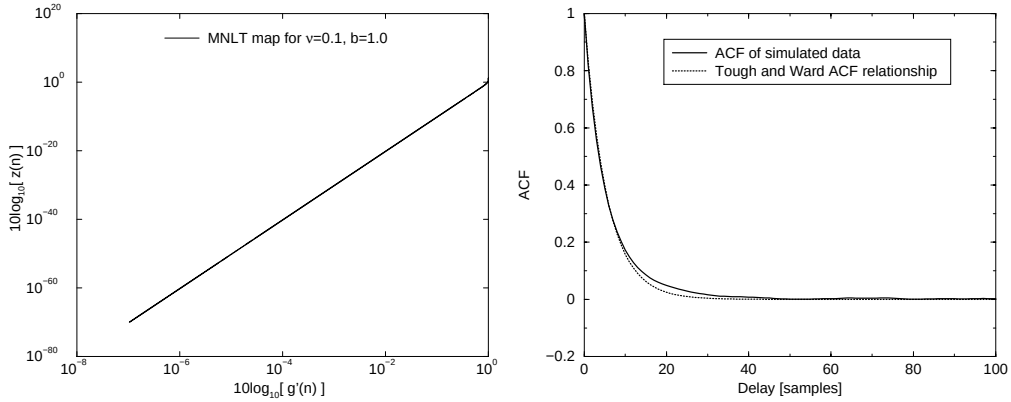
3.5.6 Simulation of correlated K-distributed data with a shape parameter ν equal to 0.1

This section presents generation details for the simulation of K-distributed data with a shape parameter value ν equal to 0.1. As discussed in Chapter 2, smaller values of shape parameter are associated with more spiky data sets. A MNLT map was produced to generate correlated gamma with a shape parameter ν equal to 0.1, and a scale parameter b equal to 1. The correlation coefficient of the IIR filter used to correlate the Gaussian sequence $\{g'(n)\}$, from which the correlated gamma sequence was obtained, was 0.9. This data was used, together with an uncorrelated Gaussian speckle component, to produce amplitude K-distributed data with a shape parameter ν equal to 0.1, and a scale parameter c equal to 0.6267.

The Maple script parameter values used to generate the MNLT map which was used to produce the gamma data were as follows: *min*=1.0e-70, *max*=15.88, *step*=1.005. This resulted in a MNLT map with 32872 values.

Figure 3.4 shows the MNLT map, and the ACF for the correlated gamma data. Figure 3.4(b) demonstrates the good fit obtained between the ACF of the correlated gamma data, with the empirical ACF derived by Tough and Ward, see equations (3.19) and (3.20).

Sample moments analysis for the correlated gamma and amplitude K-distributed data sets was carried out as follows. One million correlated gamma and amplitude K-distributed variates were generated. Then the first 10 sample moments were estimated for each data set using 100



(a) **(b)**

Figure 3.4: *Generation of correlated gamma distributed data with ν=0.1, b=1, and IIR filter correlation coefficient ρ=0.9, (a) MNLT map, (b) ACF of the correlated gamma data and Tough and Ward's empirical formula for ACF.*

successive blocks of data, each of length 10,000 samples. The mean and standard deviation were calculated for each sample moment estimate. The moments analysis for the gamma data is given in Table 3.3. The moments analysis for the amplitude K-distributed data is given in Table 3.4. A good agreement was obtained between the desired theoretical moment values and the estimated values obtained from the simulated data.

Moment	**Theoretical value**	**Estimated mean**	**Estimated standard deviation**
1	1.00e-1	1.00e-1	1.00e-2
2	1.10e-1	1.10e-1	2.00e-2
3	2.30e-1	2.20e-1	1.00e-1
4	7.20e-1	6.80e-1	6.10e-1
5	2.94e+0	2.93e+0	4.81e+0
6	1.50e+1	1.64e+1	4.37e+1
7	9.13e+1	1.14e+2	4.34e+2
8	6.49e+2	9.17e+2	4.55e+3
9	5.25e+3	8.25e+3	4.93e+4
10	4.78e+4	7.99e+4	5.46e+5

Table 3.3: *Moments analysis of correlated gamma data with ν=0.1, b=1, and IIR filter correlation coefficient ρ=0.9.*

Moment	Theoretical value	Estimated mean	Estimated standard deviation
1	1.60e-1	1.50e-1	1.00e-2
2	1.27e-1	1.24e-1	1.35e-2
3	1.79e-1	1.72e-1	2.92e-2
4	3.57e-1	3.31e-1	8.57e-2
5	9.14e-1	8.10e-1	3.14e-1
6	2.86e+0	2.37e+0	1.34e+0
7	1.06e+1	7.98e+0	6.36e+0
8	4.52e+1	2.99e+1	3.21e+1
9	2.18e+2	1.22e+2	1.69e+2
10	1.18e+3	5.33e+2	9.21e+2

Table 3.4: *Moments analysis of correlated amplitude K-distributed data with ν=0.1, c=0.6267, and IIR filter correlation coefficient ρ=0.9.*

3.6 Obtaining estimates of the speckle and gamma components of real sea clutter data for the generation of surrogate data

3.6.1 Introduction

This section discusses an approach to obtain the pdf and ACF of the gamma component, and the ACF of the speckle component, of real sea clutter data. Given these properties, surrogate K-distributed data can be generated with the same compound components as the actual sea clutter data.

3.6.2 Determining if clutter is locally Rayleigh

Before attempting to determine the properties of the compound components of sea clutter, it must first be established if the compound K-distribution is a suitable model for the clutter data. To establish this, Ward *et al.* [5] first of all considered if the clutter data was locally Rayleigh. They did this by plotting the cumulative distributions of short time sequences of amplitude sea clutter data, on "Weibull paper". This is done by making a plot of $10\log_{10}[v_t]$ versus $10\log_{10}\left[\ln\left[\frac{1}{P_{fa}}\right]\right]$, where v_t is the independent variable, and P_{fa} is the dependent variable. The variable v_t is termed the amplitude threshold value, and P_{fa} is termed the probability of false alarm[3]. The P_{fa} value is calculated for a given v_t value, by counting the number of clutter

[3]Probability of false alarm is discussed in more detail in Chapter 7.

amplitude samples that cross (*i.e.* are greater than) v_t, and dividing this by the total number of clutter samples that are compared with v_t. If the "Weibull paper" plots of the amplitude clutter short time sequences are straight lines with the same slope as the "Weibull paper" plot for Rayleigh data, then the locally Rayleigh nature of the clutter data can be said to have been demonstrated [5, 62]. Ward *et al.* demonstrated the locally Rayleigh nature of amplitude sea clutter data sets, by analysing short time sequences of length 250ms.

3.6.3 The ACF of the speckle component

If the "Weibull paper" plot of the amplitude clutter data does demonstrate that the clutter is locally Rayleigh, then the properties of the compound components of the clutter can be estimated. The average complex ACF of the speckle component $R_{g''g''}(m)$ of clutter can be estimated from the in-phase and quadrature-phase channels of the sea clutter data, as described in equation (3.14). This ACF estimation should be done on the same short time sequences that were demonstrated to be Rayleigh, using "Weibull paper" plot: an ACF estimation should be made for each short time sequence, and then these estimates should be averaged to obtain the average ACF of the speckle component. This average complex speckle ACF can be used to calculate the complex weights for the speckle correlation filter $H(w)$, as discussed in section 3.5.3.

3.6.4 The pdf of the gamma component

A sliding window of the same length as that of the short time sequences used to produce the "Weibull paper" plot (to demonstrate the locally Rayleigh nature of clutter) should be averaged[4], to produce estimates of the gamma component $z(n)$ of sea clutter as follows,

$$z(n) = \frac{1}{J}\sum_{i=1}^{J} x^2(i) \tag{3.25}$$

where there are J samples in the sliding window, and $x(i)$ is a clutter amplitude sample.

Once the gamma samples have been estimated, the method of moments (see Appendix B) can be used to obtain shape and scale parameter estimates for the gamma data. These parameters

[4]This was the approach adopted by Ward *et al.* in [5].

can then be used to produce the discrete MNLT map, discussed in section 3.5.4.

3.6.5 The ACF of the gamma component

The ACF of the gamma component of the real sea clutter data, discussed in section 3.6.4, can be estimated. This ACF can then be used to set the ACF of surrogate correlated gamma data, using Tough and Ward's MNLT method, as follows. In an ad-hoc fashion, the correlation coefficient of the IIR filter (as discussed in section 3.5.4) can be selected to produce a correlated gamma sequence (after the MNLT from correlated Gaussian to correlated gamma data) with an ACF as close to the ACF estimate of the real data as possible.

3.7 Summary

This chapter discussed the multiplicative nature of the K-distribution, and explained that this underpins the techniques discussed for the simulation of correlated K-distributed data. Techniques already presented in the literature for the simulation of correlated K-distributed data were briefly reviewed. The simulation approach of Tough and Ward was discussed in some detail, and results using this approach were presented. A methodology to obtain the pdf and ACF of the gamma component, as well as the ACF of the speckle component, of sea clutter data was presented. It was highlighted that before trying to obtain such estimates, the locally Rayleigh nature of the clutter data should first be established.

Chapter 4
Nonlinear prediction of chaotic signals

4.1 Introduction

In [99] Scott and Mulgrew compared a global linear predictor with a global nonlinear predictor on chaotic time series from two different attractors (Duffing and Lorenz), and in both cases the nonlinear predictor was shown to perform better than the linear predictor. This result is to be expected due to the nonlinear evolution in state space of a chaotic time series, as discussed in Chapter 2. The term global refers to the fact that one predictor was used on the chaotic time series for all parts of the corresponding attractor. Casdagli [100] demonstrated that a locally linear predictor could achieve a similar performance to that of a global nonlinear predictor, on a chaotic time series from the Lorenz attractor. The locally linear technique involved fitting to one small part of the attractor at a time, effectively requiring different linear predictors for different parts of the attractor. In other words one nonlinear prediction problem was broken down into a number of smaller linear prediction problems.

The prediction of chaotic time series using global linear and nonlinear techniques is further explored in this chapter. This is done using the Volterra series filter (VSF) [23, 101–104] and the radial basis function network (RBFN) [24, 105–114]. These two nonlinear models can be regarded as *linear in their parameters*, which means that they can be trained using fast supervised linear techniques. A linear predictor is used to obtain a performance benchmark for the nonlinear predictors.

This chapter is structured as follows. In section 4.2 the general problem of nonlinear prediction is outlined. This is followed in sections 4.3 and 4.4 by a discussion of the structure and training of VSF's and RBFN's, respectively. In section 4.5 the problem of over-fitting is highlighted, and a simple method to avoid such a problem is described. Additionally, two measures for assessing the performance of a nonlinear predictor are discussed: normalised mean square error (NMSE), and recursive prediction. In section 4.6 a well known chaotic signal, Logistic map data, is used to investigate the performance of a RBFN predictor (RBFNP) using two different centres selection algorithms, and two different Gaussian RBFN architectures. A cubic VSF predictor

(VSFP) is used as a comparison for the RBFNP's. In section 4.7 another well known chaotic signal, Lorenz data, is also used to compare the performance of RBFNP's using the same centres selection algorithms and network architectures used on the Logistic map data. Moreover, noisy Lorenz data is used to investigate the ability of a RBFNP to capture the underlying dynamics of a chaotic signal. A chapter summary is presented in section 4.8.

4.2 Nonlinear prediction

Given a vector $\mathbf{x}(n)$, from a time series $\{x(n)\}$, which has an embedding dimension N and an embedding delay τ, *i.e.*

$$\mathbf{x}(n) = [x(n), x(n-\tau), ..., x(n-(N-1)\tau)]^T \tag{4.1}$$

an estimate $\hat{x}(n+1)$ of the next data sample $x(n+1)$ is formed by constructing a nonlinear predictor function $f()$ where,

$$\hat{x}(n+1) = f(\mathbf{x}(n)) \tag{4.2}$$

Equation (4.2) is for 1-step ahead prediction, but this could be generalised for K-step ahead prediction, *i.e.*

$$\hat{x}(n+K) = f_K(\mathbf{x}(n)) \tag{4.3}$$

where $f_K(\mathbf{x}(n))$ would, in general, represent a different predictor function for each value of K.

4.3 Volterra series filter

4.3.1 Structure

A Volterra series filter (VSF) is a polynomial model of nonlinearity [23]. It is a feed-forward filter: there is no feedback path. A nonlinear filter cannot be characterised by its unit impulse response [4], therefore it is very difficult to design a stable system with feedback. The Volterra series (VS) equation, equation (4.4), can be used to find the predictor function discussed in section 4.2. For an input $\mathbf{x}(n)$, and output $f(\mathbf{x}(n)$, of a discrete-time and causal nonlinear system, a VS expansion can be written [23] as,

$$\begin{aligned} f(\mathbf{x}(n)) &= h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) \\ &+ \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1,m_2)x(n-m_1)x(n-m_2) + ... \\ &+ \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty}\cdots\sum_{m_L=m_0}^{\infty} h_2(m_1,m_2,...,m_L)\times x(n-m_1)...x(n-m_L) \\ &+ ... \end{aligned} \tag{4.4}$$

where $h_L(m_1, m_2, ..., m_L)$ is known as the L^{th} order Volterra kernel of the system. Without loss of generality it may be assumed that the Volterra kernels are symmetric, *i.e.* $h_L(m_1, m_2, ..., m_L)$ is left unchanged for any of the possible $L!$ permutations of the indices $m_1, m_2, ..., m_L$. Also, in practise a truncated VS expansion must be used, so that equation (4.4) may be written as,

$$\begin{aligned} f(\mathbf{x}(n)) &= \sum_{m_1=0}^{N-1} h_1(m_1)x(n-m_1) \\ &+ \sum_{m_1=0}^{N-1}\sum_{m_2=m_1}^{N-1} h_2(m_1,m_2)x(n-m_1)x(n-m_2) + ... \\ &+ \sum_{m_1=0}^{N-1}\cdots\sum_{m_L=m_{L-1}}^{N-1} h_2(m_1,m_2,...,m_L)\times x(n-m_1)...x(n-m_L) \end{aligned} \tag{4.5}$$

The truncated VS expansion given above is for a zero mean process (*i.e.* $h_0 = 0$). N is known

as the degree of the VS (which is equivalent to the embedding dimension of $\mathbf{x}(n)$). L is known as the order of the VS.

A disadvantage of the VS nonlinear model is that the VS complexity grows rapidly with N and L. For the work in this thesis, a VSF of up to order 3 was only considered. Below is shown a breakdown of the number of terms in each section (linear, quadratic, and cubic) of the VSF,

$$\begin{array}{lll} \text{Linear} & (L=1): & N \\ \text{Quadratic} & (L=2): & \frac{(N+1)!}{2!(N-1)!} \\ \text{Cubic} & (L=3): & \frac{(N+2)!}{3!(N-1)!} \end{array}$$

where N is the embedding dimension of the input signal.

It should be noted that for the work reported in this thesis, a cubic VSF refers to a VSF with a linear section, a quadratic section, and a cubic section. Likewise, a quadratic VSF consists of both a linear section and a quadratic section. A graphical example of a VSF is given in Figure 4.1. This shows a quadratic (L=2) VSF, of degree N=3, which operates on an input vector of embedding dimension N=3, and embedding delay τ=1.

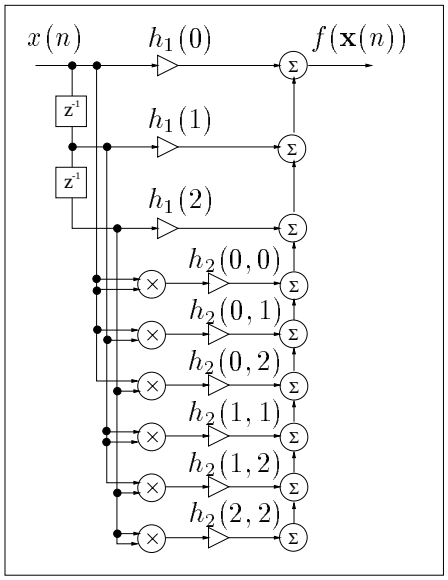


Figure 4.1: *Quadratic VSF, L=2, N=3.*

4.3.2 Training

To demonstrate that the VSF is linear in its parameters, equation (4.5) is re-written as a vector inner product,

$$f(\mathbf{x}(n)) = \mathbf{h}^T \mathbf{x}_{expanded}(n) \tag{4.6}$$

where,

$$\begin{aligned} \mathbf{x}^T_{expanded}(n) &= [x(n), x(n-1), \ldots, x(n-N+1), \ldots, x(n)x(n)\ldots x(n), \ldots \\ &\quad x(n-N+1)x(n-N+1)\ldots x(n-N+1)] \end{aligned}$$

$$\begin{aligned} \mathbf{h}^T &= [h_1(0), h_1(1), \ldots, h_1(N-1), \ldots h_L(0,0,\ldots 0), h_L(N-1, N-1, \ldots \\ &\quad N-1)] \end{aligned}$$

The weight vector $\mathbf{h}^T$ can be estimated using a least squares algorithm. One of the most well known adaptive least squares algorithm is the least mean squares (LMS) algorithm. However, it was decided not to use this technique as Mathews [23] observed that the nonlinear expansion of the input vector $\mathbf{x}(n)$ will cause the eigenvalue spread of the ACF matrix to increase, even when the input signal is white. Furthermore, it is known that the LMS algorithm convergence speed decreases when the eigenvalue spread of the ACF matrix is increased [18]. Instead, a numerically robust, block least squares algorithm known as the Householder transform [115, 116] was used, see Appendix D, to train the Volterra kernels for the work in this thesis. This technique avoids estimating the inverse of the ACF matrix directly. Figure 4.2 shows a block diagram of how a VSF of arbitrary order and degree can be trained to operate as a K-step ahead NLP: a VSF predictor (VSFP).

Finally, it should be noted that the VS expansion does not perform well when there are discontinuities [23]. This is because the VS expansion is a Taylor series expansion with memory which only works well when the function it attempts to model is smooth, in the sense that it is at least once differentiable [4].

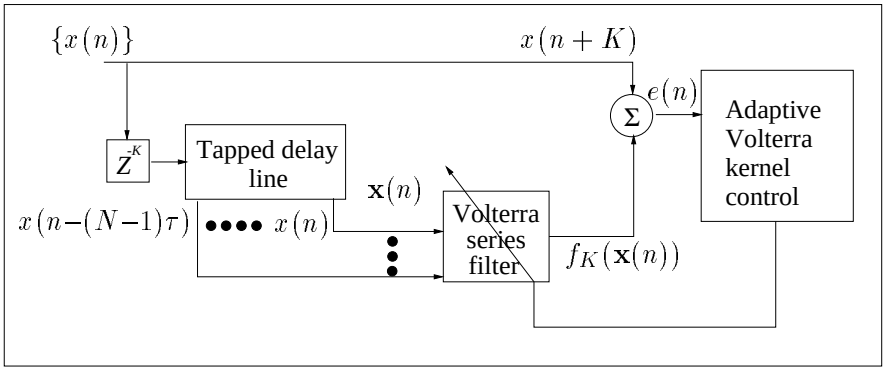


Figure 4.2: *Block diagram of the training of a VSF as a K-step ahead NLP. $\mathbf{x}(n)$ is the input vector to the VSF. This input vector has, and the VSF is also said to have, an embedding dimension of N and an embedding delay of τ. The VS kernels are adapted using a least squares algorithm to minimise the difference between the desired value $x(n+K)$ and the output of the Volterra series filter, $f_K(\mathbf{x}(n))$* (i.e. *minimise $x(n+K) - \hat{x}(n+K)$*).

4.4 Radial basis function network

4.4.1 Introduction

The radial basis function network (RBFN) [24, 105–114], and the multi-layer perceptron (MLP) [24, 114], are feed-forward networks that can be used to implement a NLP. Both have been used for the nonlinear prediction of sea clutter [12, 18], which is the subject of Chapter 5. However, as Haykin [24] points out, there always exists a RBFN capable of accurately mimicking a specified MLP or vice versa. So, if data is nonlinearly predictable, it would be expected that both a MLP and a RBFN would be able to exploit this predictability, and hence there is no need to implement both networks, only one. For this work a RBFN has been chosen in preference to a MLP for the following reason [114]. The training time of the MLP suffers from the training process being highly nonlinear. However, after choosing suitable basis function parameters (discussed in section 4.4.3), the RBFN can be trained using fast linear supervised methods (discussed in section 4.4.4).

4.4.2 Structure

As discussed above a RBFN can be used to find the predictor function presented in section 4.2. The structure of a RBFN is shown in Figure 4.3.

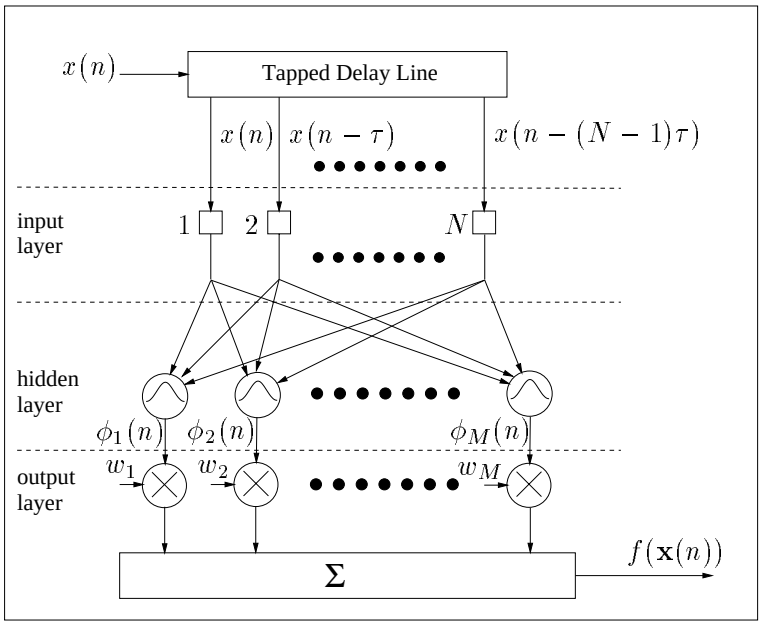


Figure 4.3: *RBFN with N source nodes in the input layer, M Gaussian kernels $\phi_i(n), i = 1, 2, ..., M$, in the hidden layer, and M weights $w_i, i = 1, 2, ..., M$, in the output layer. The input vector has, and therefore the RBFN is said to also have, an embedding dimension N and embedding delay τ.*

The input layer consists of N source nodes, which may be formed by using a tapped delay line to represent an embedding of an observed time series. The hidden layer nonlinearly expands the input signal. The hidden layer is shown in Figure 4.3 consisting of M units. The units of the hidden layer are known as the kernels of the RBFN, and each kernel output $\phi_i(n)$ is multiplied by a corresponding weight w_i. The weighted kernel outputs are then summed to produce the output of the RBFN.

A RBFN can be used to find the predictor function discussed in section 4.2. For an input vector $\mathbf{x}(n)$, equation (4.1), the corresponding output $f(\mathbf{x}(n))$ of a RBFN with N source nodes and M kernels is written as,

$$f(\mathbf{x}(n)) = \sum_{i=1}^{M} w_i \phi_i(n) \tag{4.7}$$

The most common nonlinear kernel function used in RBFN's is the Gaussian function,

$$\phi_i(n) = \exp\left[\frac{-\|\mathbf{x}(n) - \mathbf{c}_i\|^2}{2\sigma_i^2}\right], \quad i = 1, 2, ...M \tag{4.8}$$

where $\phi_i(n)$ is the output of the i^{th} kernel, $\|.\|$ is the Euclidean distance measure,

$$\|\mathbf{x}_i - \mathbf{x}_j\| = \left[\sum_{n=1}^{N}(x_{in} - x_{jn})^2\right]^{\frac{1}{2}} \tag{4.9}$$

$\mathbf{c}_i$ is the position of the i^{th} kernel's centre in N-dimensional space, and σ_i is known as the width of the i^{th} kernel. The type of nonlinear kernel function is not crucial for the performance of the RBFN [109]. Other choices for the nonlinear function include the thin-plate-spline and the multi-quadratic functions [109].

4.4.3 Normalised Gaussian kernels

Two different Gaussian RBFN architectures were considered. Firstly, a RBFN which used the Gaussian kernel function in equation (4.8) was considered. This will be referred to from now on as the un-normalised RBFN or UNRBFN. Secondly, a RBFN which used the normalised Gaussian kernel function [117–123], see equation (4.10), was considered. This will be referred to from now on as the normalised RBFN or NRBFN. The normalised Gaussian kernel function is defined below,

$$\phi_i(n) = \frac{\exp\left[\frac{-\|\mathbf{x}(n)-\mathbf{c}_i\|^2}{2\sigma_i^2}\right]}{\sum_{j=1}^{M}\exp\left[\frac{-\|\mathbf{x}(n)-\mathbf{c}_j\|^2}{2\sigma_j^2}\right]} \tag{4.10}$$

where $\phi_i(n)$ is the i^{th} normalised kernel output. This is very similar to equation (4.8), except here the kernel output is divided by the sum of all the kernel outputs. Therefore, the outputs of all the kernels add up to one.

4.4.4 Selection of basis function parameters

As mentioned in section 4.1, the free parameters $\mathbf{c}_i$, σ_i, and w_i, for $i = 1, 2, ..., M$, of the RBFN can be chosen in such a way so as to avoid a time consuming nonlinear optimisation selection strategy. This usually involves selecting the basis function parameters $\mathbf{c}_i$ and σ_i using an un-supervised method, and then using a fast linear least squares supervised method to train the output layer weights w_i. This section deals with the selection of the basis function parameters: the kernel centres $\mathbf{c}_i$ and the kernel widths σ_i. Two un-supervised methods will be discussed for the selection of $\mathbf{c}_i$ from the training data. One method is discussed for the selection of σ_i.

Centres selection method I: randomly selected centres (RSC)

The randomly selected centres (RSC) technique involves choosing centres from the training data at random. A uniform random number generator [87] (RNG) is used to pick points at random from the training data. These points are used as the starting elements of the centres. For example if the RBFN has an embedding dimension of N, and an embedding delay of 1 sample, then a starting element is picked, along with the next $N - 1$ successive data points, to obtain centres in N-dimensional space. Care must be taken to ensure that no centre is picked more than once, to avoid ill-conditioning of the data matrix sent to the Householder transform. This selection technique was implemented so that, given two RBFN's with the same embedding dimension, embedding delay, training length, and training data set, but one with M_1 kernels, and the other with M_2 kernels, where $M_2 > M_1$, the set of M_1 centres is a subset of the M_2 centres.

Centres selection method II: the optimal k-means clustering algorithm

The optimal adaptive k-means (OAKM) algorithm [124] is used to partition the input N-dimensional space of the RBFN into M clusters. The centres of the kernels are positioned onto the centres of these clusters. The OAKM algorithm achieves an optimal or near-optimal partition of the input space independent of the initial centre locations (which the algorithm requires) [125]. The RSC method was used to pick the centres for the initialisation of the OAKM technique.

Kernel width selection

For both of the above centres selection methods the same technique was used to find the kernel widths $\sigma_i, i = 1, 2, ..., M$. In fact, a universal width was used (*i.e.* the same width for each

kernel): $\sigma_i = \sigma, i = 1, 2, ..., M$. The following equation was used to calculate the width σ,

$$\sigma = \frac{d_{me}}{\sqrt{2M}} \tag{4.11}$$

where d_{me} is the maximum Euclidean distance between any 2 centres, and M is the number of kernels. Such a choice for σ ensures that the Gaussian kernel functions are neither too peaked nor too flat [24].

4.4.5 Training the output layer weights of the RBFN

Once the centres and widths of the M kernel functions have been selected, the M linear output layer weights $w_i, i = 1, 2, .., M$, can be trained using a supervised linear least squares technique. The least squares approach involves solving equation (4.12),

$$\Phi(n)\mathbf{w} = \mathbf{x}(n + K) \tag{4.12}$$

where,

$$\Phi(n) = \begin{bmatrix} \phi_1(n) & \phi_2(n) & \cdots & \phi_M(n) \\ \phi_1(n+1) & \phi_2(n+1) & \cdots & \phi_M(n+1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(n+(A-1)) & \phi_2(n+(A-1)) & \cdots & \phi_M(n+(A-1)) \end{bmatrix} \tag{4.13}$$

is a $A \times M$ matrix of kernel outputs $\phi_i(n)$, M is the number of kernels in the hidden layer of the RBFN, and $A \geq M$. $\mathbf{w}$ is the $M \times 1$ vector consisting of the RBFN's output layer weights $w_i, i = 1, 2, ..., M$, and $\mathbf{x}(n + K)$ is the $M \times 1$ vector consisting of the *desired* K-step ahead RBFN output samples. The training of a RBFN with an embedding dimension of N and embedding delay τ, is illustrated in Figure 4.4.

The nonlinear layer of the RBFN causes an increase in the eigenvalue spread of the ACF mat-

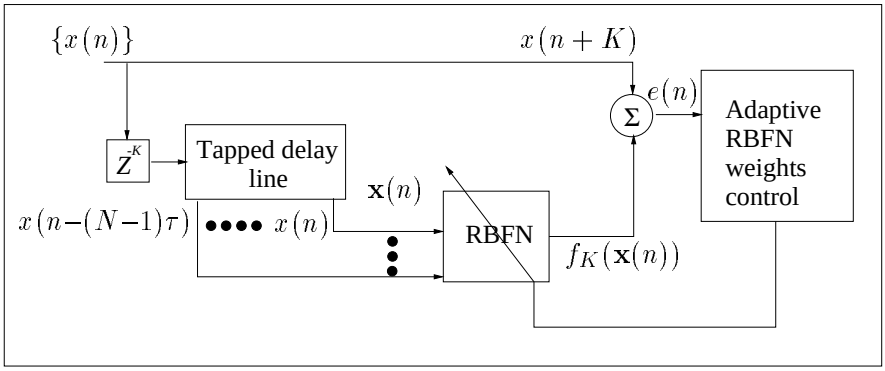


Figure 4.4: *Block diagram of the training of a RBFN as a K-step ahead NLP. $\mathbf{x}(n)$ is the input vector to the RBFN. This input vector has, and the RBFN is also said to have, an embedding dimension of N and an embedding delay of τ. The RBFN weights are adapted using a least squares algorithm to minimise the difference between the desired value $x(n+K)$ and the output of the RBFN, $f_K(\mathbf{x}(n))$* (i.e. *minimise $x(n+K) - \hat{x}(n+K)$*).

rix [4]. Therefore, as was used for the training of the kernels of a VSF, the numerically robust Householder transform was used to train the output layer weights $w_i, i = 1, 2, ..., M$, of the RBFN. Strauch [4] pointed out that if the basis function parameters are not selected carefully, the resulting data matrix, (formed from the outputs of the kernels), will often be ill-conditioned[1]. This ill-conditioning can be caused by too small a width value, or a kernel centre placed in a sparsely populated area of the input space. The selection techniques for the basis functions described in section 4.4.4 were chosen with the aim of avoiding such ill-conditioning.

This ill-conditioning can lead to numerical problems. To avoid such problems, a technique known as singular value decomposition (SVD) [87] can be used to train the weights, instead of the Householder transform. Strauch [4] recommended the use of SVD when the data matrix is ill-conditioned, and when the data matrix is well-conditioned to use the Householder transform.

4.4.6 Regularisation theory

In an article by Haykin and Principe [29], regularisation theory was applied to RBFN's in order to reconstruct the dynamics of a chaotic signal in noise. An UNRBFN was shown to be unable

[1] If the condition number ($\frac{max_{eigenvalue}}{min_{eigenvalue}}$) of a matrix is too large, that is, if the reciprocal of the condition number approaches the machine's floating-point precision.

to recursively[2] predict (and therefore capture the dynamics of) Lorenz data corrupted by white Gaussian noise, where the signal to noise ratio (SNR) was 25dB. However, an UNRBFN which used regularised training, or a regularised UNRBFN (RUNRBFN), was able to recursively predict the noisy Lorenz signal. For this reason, the use of RUNRBFN's was investigated.

The basic idea of regularisation is to stabilise the solution to the approximation problem, (*i.e.* the approximation of the predictor function), by making use of prior information about the input-output (*i.e.* RUNRBFN) mapping and to thereby make an ill-posed[3] problem into a well-posed problem [24].

Unlike the usual or un-regularised training (section 4.4.5) of the output layer weights, which works for both UNRBFN's and NRBFN's, regularised training [24, 106] has only been theoretically justified to work for an UNRBFN [106]. A further restriction of regularised training is that the nonlinear kernel function used must produce an output that is positive definite for all input vectors [24]. The Gaussian kernel function, equation (4.8), satisfies this condition.

The regularised least squares approach involves solving the following equation,

$$(\Phi(n) + \lambda \mathbf{I})\, \mathbf{w} = \mathbf{x}(n+K) \tag{4.14}$$

where $\Phi(n)$ is a $A \times M$ matrix, as given above in equation (4.13), M is the number of kernels in the hidden layer of the RBFN, with $A \geq M$, $\mathbf{w}$ is the $M \times 1$ vector consisting of the RBFN's output layer weights $w_i, i = 1, 2, ..., M$, $\mathbf{x}(n+K)$ is the $M \times 1$ vector consisting of the *desired* K-step ahead RBFN output samples, $\mathbf{I}$ is the $M \times M$ identity matrix, and λ is known as the regularisation parameter . Unfortunately, this equation only holds if there are as many kernels in the RBFN's hidden layer as there are input vectors in the training data file, which is used to train the RBFN [24]. To avoid this restriction, and to use fewer kernels than input training vectors, the following equation can be solved instead of equation (4.14), using a least squares approach, to produce an approximately [24] regularised solution,

[2] Recursive prediction is discussed in section 4.5.2.

[3] the problem of learning a smooth mapping from input-output pair examples is ill-posed in the sense that the information in the data is not sufficient to reconstruct, uniquely, the mapping in regions where data are not available.

$$\left(\Phi^T(n)\Phi(n) + \lambda\Phi_o(n)\right)\mathbf{w} = \Phi^T(n)\mathbf{x}(n+K) \tag{4.15}$$

where Φ_o is a symmetric $M \times M$ matrix defined by,

$$\Phi_o = \begin{bmatrix} \phi(\mathbf{c}_1,\mathbf{c}_1) & \phi(\mathbf{c}_1,\mathbf{c}_2) & \cdots & \phi(\mathbf{c}_1,\mathbf{c}_M) \\ \phi(\mathbf{c}_2,\mathbf{c}_1) & \phi(\mathbf{c}_2,\mathbf{c}_2) & \cdots & \phi(\mathbf{c}_2,\mathbf{c}_M) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\mathbf{c}_M,\mathbf{c}_1) & \phi(\mathbf{c}_M,\mathbf{c}_2) & \cdots & \phi(\mathbf{c}_M,\mathbf{c}_M) \end{bmatrix} \tag{4.16}$$

where,

$$\phi(\mathbf{c}_i,\mathbf{c}_j) = \exp\left[\frac{-\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2\sigma_i^2}\right];\ \ i,j = 1,2,...M \tag{4.17}$$

and σ is the universal width, as defined in equation (4.11). The terms Φ, $\mathbf{w}$, $\mathbf{x}(n+K)$, and λ are as described above for equation (4.14).

For convenience, using equation (4.14) to obtain RBFN weights, will be referred to as full regularisation, and using equation (4.15) will be referred to as approximate regularisation.

4.5 Over-fitting and prediction performance measurement

Before presenting any prediction results, the subjects of over-fitting and prediction performance measurement must be discussed.

4.5.1 Over-fitting

Over-fitting [114] occurs when a model, (*i.e.* RBFN or VSF), is too complex and fits to spurious quirks (*i.e.* noise) in the data. Another way of viewing this is to say that the model does not provide a smooth mapping between the input to the model, and the desired output. This means

that the model will perform well on training data, but will perform less well on data it has not seen, and will therefore have poor generalisation properties [114].

To avoid over-fitting, a technique known as early stopping [114] can be used. This technique simply involves splitting the data into training and non-training data sets. For example, with the prediction problem, the training error (*i.e.* the desired output minus the predictor's output, for the training data set), is expected to continually decrease with time (and training!), until it hits a noise floor. However the non-training error is expected to at first decrease with time, but then eventually increase again, as the model (the predictor) starts to over-fit to the training data. By monitoring the predictor's performance on non-training data, the problem of over-fitting can be avoided: if the non-training error starts to increase, then stop the training.

For the work in this thesis the data was divided into three equal sets of length Y samples in each: a training set (the first Y samples), and 2 non-training sets; a testing set (the next Y samples), and a validation set (the next again Y samples).

4.5.2 Prediction performance measurement

Normalised mean square error (NMSE)

The prediction performance measure that has been used for this thesis is the normalised mean square error (NMSE), which is defined in equation (4.18), for the general K step-ahead prediction problem.

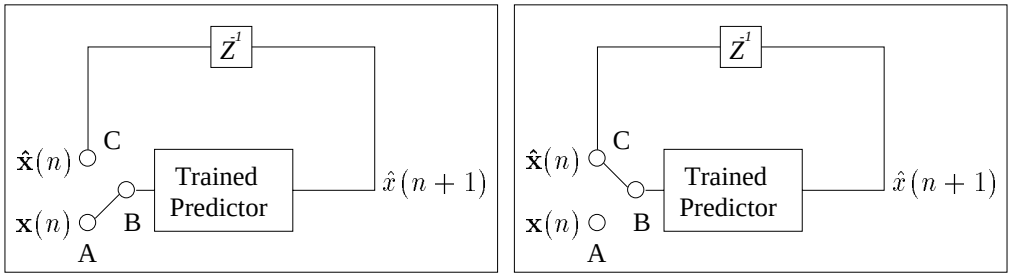
$$\mathrm{NMSE} = 10\log_{10}\left(\frac{1}{\sigma_x^2 Y}\sum_{n=1}^{Y}\left(x(n+K) - \hat{x}(n+K)\right)^2\right) \tag{4.18}$$

Recursive prediction

In addition to the NMSE performance measure, another form of prediction performance measurement, known as recursive prediction [29] was used. Recursive prediction is performed by first of all training a predictor to obtain the mapping in equation (4.2), *i.e.*

$$\hat{x}(n+1) = f(\mathbf{x}(n))$$

Then the trained predictor is given one input vector from the available data. From then on the output of the predictor is fed back to its input, and the system becomes autonomous. This is illustrated in Figure 4.5.



(a) **(b)**

Figure 4.5: *Recursive prediction. Recursive prediction is achieved as follows. As shown in (a), close switch AB, open switch BC, and initialise the trained predictor with a vector from the input data, $\mathbf{x}(n)$, to obtain an output sample, $\hat{x}(n+1)$, from the predictor. This output is the predictor's estimate of the next sample, $x(n+1)$. Next, as shown in (b), open switch AB and close switch BC. Update the vector $\mathbf{x}(n)$ with the 1-step ahead (or next sample) estimate, $\hat{x}(n+1)$, to form the next input vector $\hat{\mathbf{x}}(n)$. Keep BC closed and AB open, and continue feeding the predictor's output back to update the input vector $\hat{\mathbf{x}}(n)$.*

Haykin and Principe [29] suggested using recursive prediction as a pragmatic approach for testing how well a 1-step ahead predictor had managed to model the underlying dynamics of a chaotic signal. If the predictor is successful at modelling the underlying dynamics of the chaotic signal, then the predictor's output, in recursive prediction mode, should satisfy the two conditions [29] listed below.

1. Short term behaviour: once initialisation is completed, the reconstructed time series $\{\hat{x}(n)\}$ in Figure 4.5 should closely follow the original time series $\{x(n)\}$, for a period of time approximately equal to the prediction horizon of the signal.

2. Long term behaviour: The dynamic invariants, (such as the correlation dimension), computed from the reconstructed time series $\{\hat{x}(n)\}$ should closely match the corresponding ones from the original time series, $\{x(n)\}$. This is because these dynamic invariants measure global properties of a chaotic signal.

4.6 Prediction of Logistic map data

4.6.1 Logistic map

The Logistic map [24, 105] is a chaotic time series whose dynamics are governed by the difference equation,

$$x(n) = 4x(n-1)[1 - x(n-1)] \tag{4.19}$$

This is a first order nonlinear process where only the previous sample $x(n-1)$ determines the value of the present sample $x(n)$. The Logistic map is known to be chaotic on the interval [0,1].

Figure 4.6 shows a plot of the Logistic map in 2-dimensional state space; this plot is known as the attractor of the Logistic map.

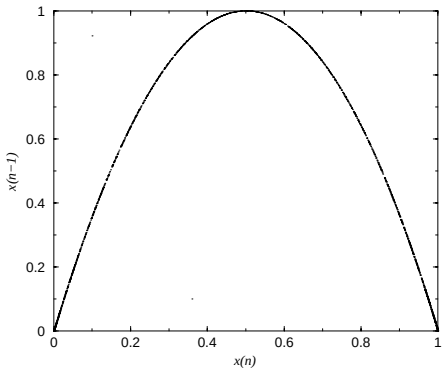


Figure 4.6: *Plot of the Logistic map attractor using 2000 data points.*

The correlation dimension estimate for the Logistic map was evaluated using the method of maximum likelihood, which was discussed in Chapter 2. The estimate for the maximum likeli-

hood correlation dimension D_{ML} was 1.07. A plot of the mutual information for the Logistic map data is given in Figure 4.7.

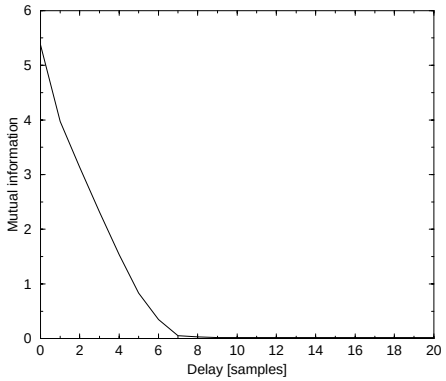


Figure 4.7: *Plot of the mutual information for the Logistic map data.*

1-step ahead prediction of the Logistic map data was carried out using linear, RBFN, and VSF predictors. This prediction analysis is now discussed.

4.6.2 Linear prediction

Prediction results are shown for a 25 tap LP [116] in Figure 4.8. The taps of the LP were trained using the Householder transform. It should be noted that, unless otherwise stated, the delay between samples for a LP is just 1 sample, and that the prediction step is also 1 sample, in the remainder of this thesis.

Figure 4.8 demonstrates that the Logistic map data is not linearly predictable. The reason for this becomes clear when the second order statistics of the Logistic map are inspected. Figure 4.9 shows a plot of the power spectral density (PSD) for the Logistic map data, formed using 65536 samples, as well as a plot of the ACF for the Logistic map data, which was also formed using 65536 samples.

From Figure 4.9 it is clear that the Logistic map spectrum is white. Therefore the Logistic map is linearly unpredictable in the sense that the value of the Logistic map at time n is uncorrelated with all past values of the process, up to and including time $n-1$, (as well as being uncorrelated with all future values of the process).

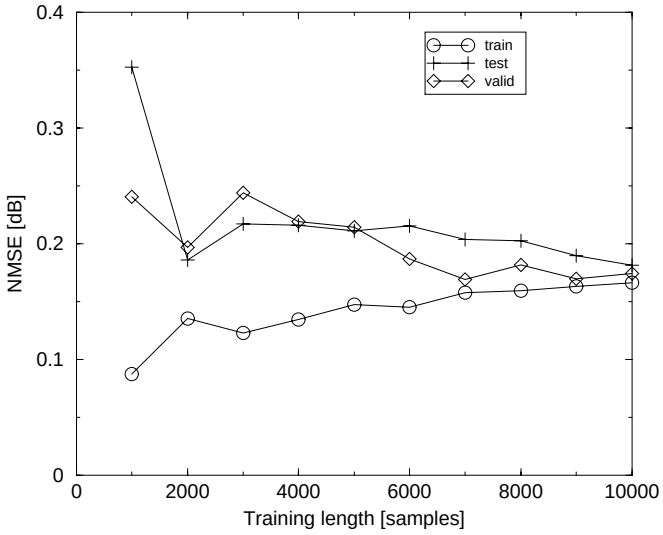
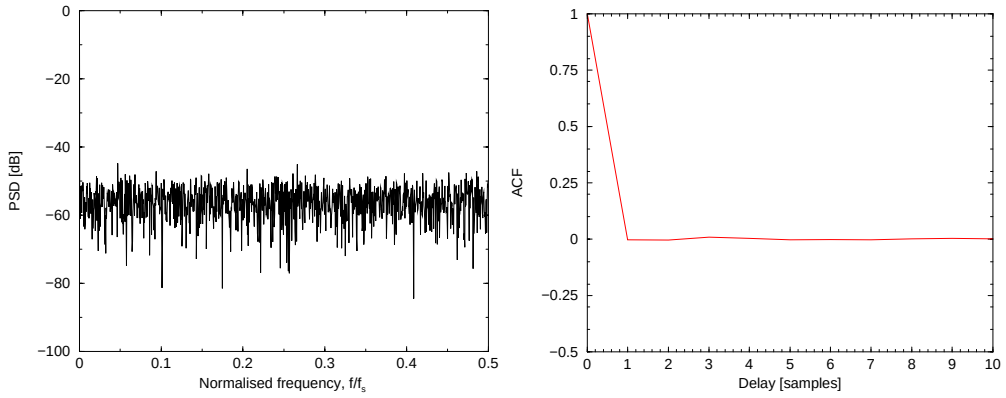


Figure 4.8: *Linear prediction of Logistic map data. Results are shown for training (train), testing (test), and validation (valid) data sets, for a 25 tap LP which was trained using the Householder transform.*



(a) **(b)**

Figure 4.9: *Second order statistics of the Logistic map, estimated using 65536 samples: (a) PSD, (b) ACF.*

4.6.3 Nonlinear prediction using RBFN's

The Logistic map, in one and two dimensional space, will be used to highlight some issues associated with choosing appropriate kernel centres for a RBFN predictor (RBFNP). The performance of NRBFN predictors (NRBFNP's) and UNRBFN predictors (UNRBFNP's) will be reported. Previous work on the prediction of Logistic map data using RBFN's is reported in [105].

RBFN prediction results are presented in Figure 4.10. Figure 4.10(a) shows results for a NRBFNP which was trained using the RSC method (section 4.4.4) to select the centre positions for the NRBFN's kernels. Figure 4.10(b) shows results for an UNRBFNP which was trained using the RSC method to pick centres. Figure 4.10(c) shows results for a NRBFNP which was trained using the OAKM algorithm (section 4.4.4) to select kernel centre positions. Finally, Figure 4.10(d) shows results for an UNRBFNP which was trained using the OAKM method to pick centres. It should be noted that for the RBFN prediction results in Figure 4.10, a different set of centres was selected for each different training length: this aspect of the RSC centres selection technique was discussed in section 4.4.4. For a given training length and centres selection technique, the same set of centres was used by both the NRBFNP and the UNRBFNP. Figure 4.10 shows results for two different cases of embedding dimension N (*i.e.* size of the RBFN input layer): N=1, and N=2. The reason for presenting results for two different cases of embedding dimension is to illustrate some aspects associated with the selection of basis function parameters, which will be discussed below. To facilitate the discussion of the results in Figure 4.10, a plot of the centres used by the RBFNP's to obtain the results in Figure 4.10 are presented in Figure 4.11, for the RBFNP's with input layer size N=1, and in Figure 4.12 for the RBFNP's with input layer size N=2. The kernel widths corresponding to the centres plotted in Figure 4.11 are given in Table 4.1. The kernel widths corresponding to the centres in Figure 4.12 are given in Table 4.2.

There are several interesting points to be observed from the prediction results in Figure 4.10. These points will now be highlighted.

NRBF vs UNRBF

Before discussing the results in Figure 4.10, it should be pointed out that following the discussion in Chapter 2, it might seem sensible to choose an embedding dimension for a RBFNP according to Takens' embedding delay theorem. However, it should be pointed out that Takens'

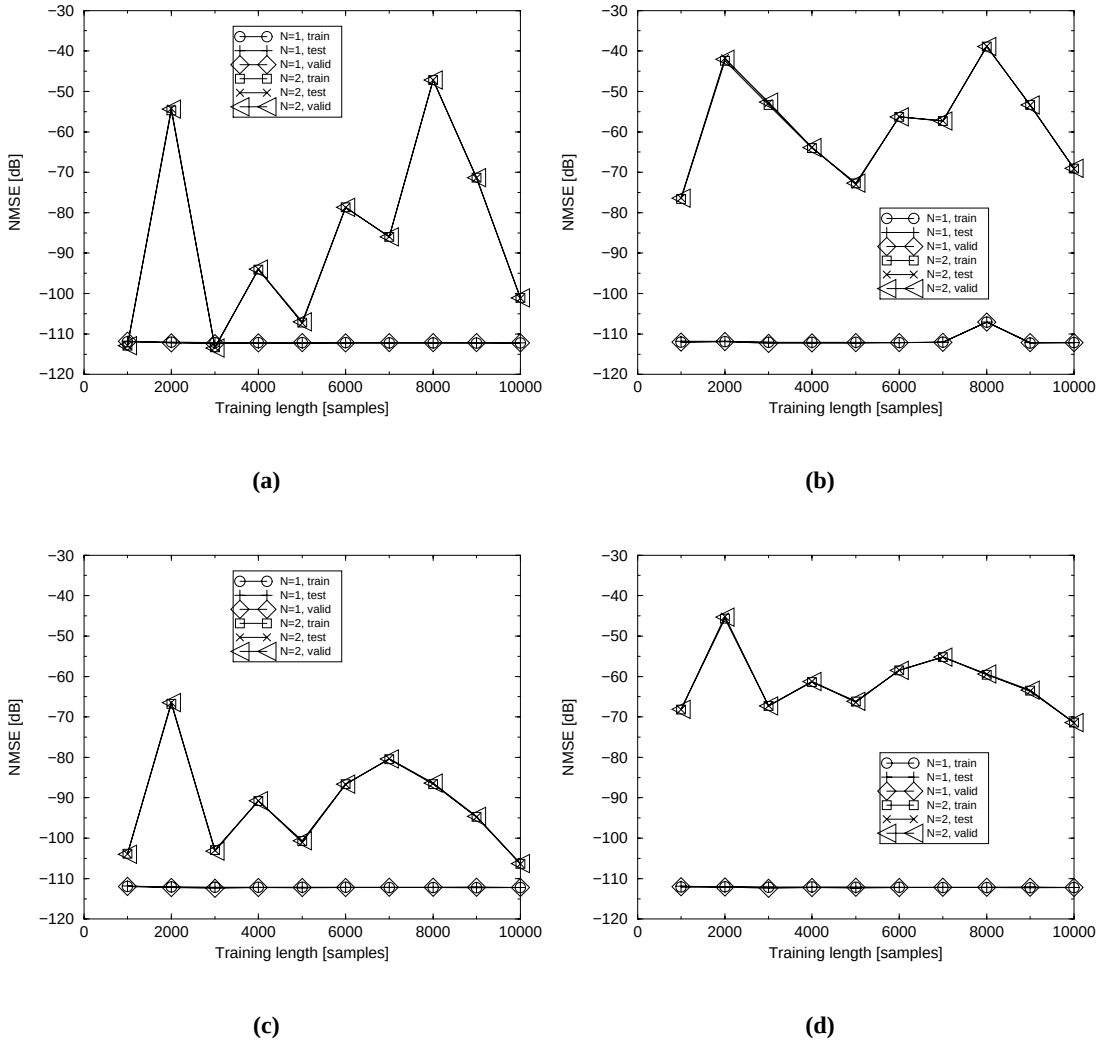


Figure 4.10: *1-step ahead nonlinear prediction of Logistic map data using (a) a NRBFNP with centres selected using the RSC method, (b) an UNRBFNP with centres selected using the RSC method, (c) a NRBFNP with centres selected using OAKM method, and (d) an UNRBFNP with centres selected using the OAKM method. NMSE results are shown for RBFNP's for 2 cases of embedding dimension (N=1,2), with an embedding delay of 1 sample and 25 kernels used for each case of embedding dimension. Results are shown for training (train), testing (test), and validation (valid) data sets.*

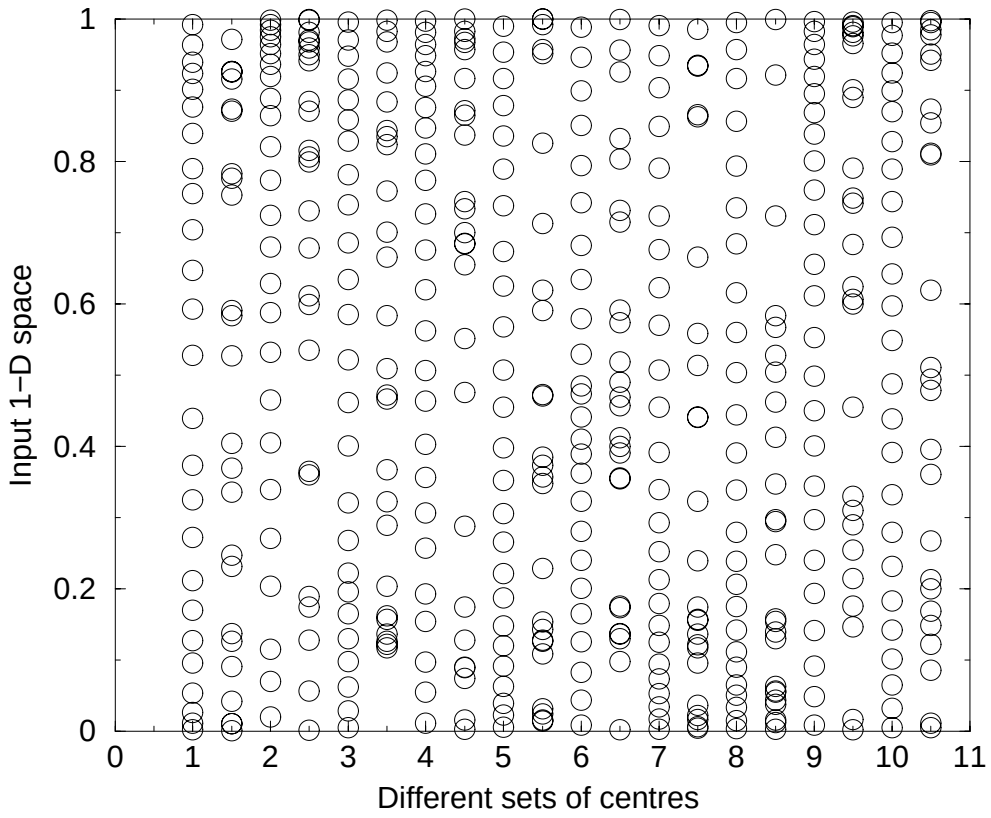
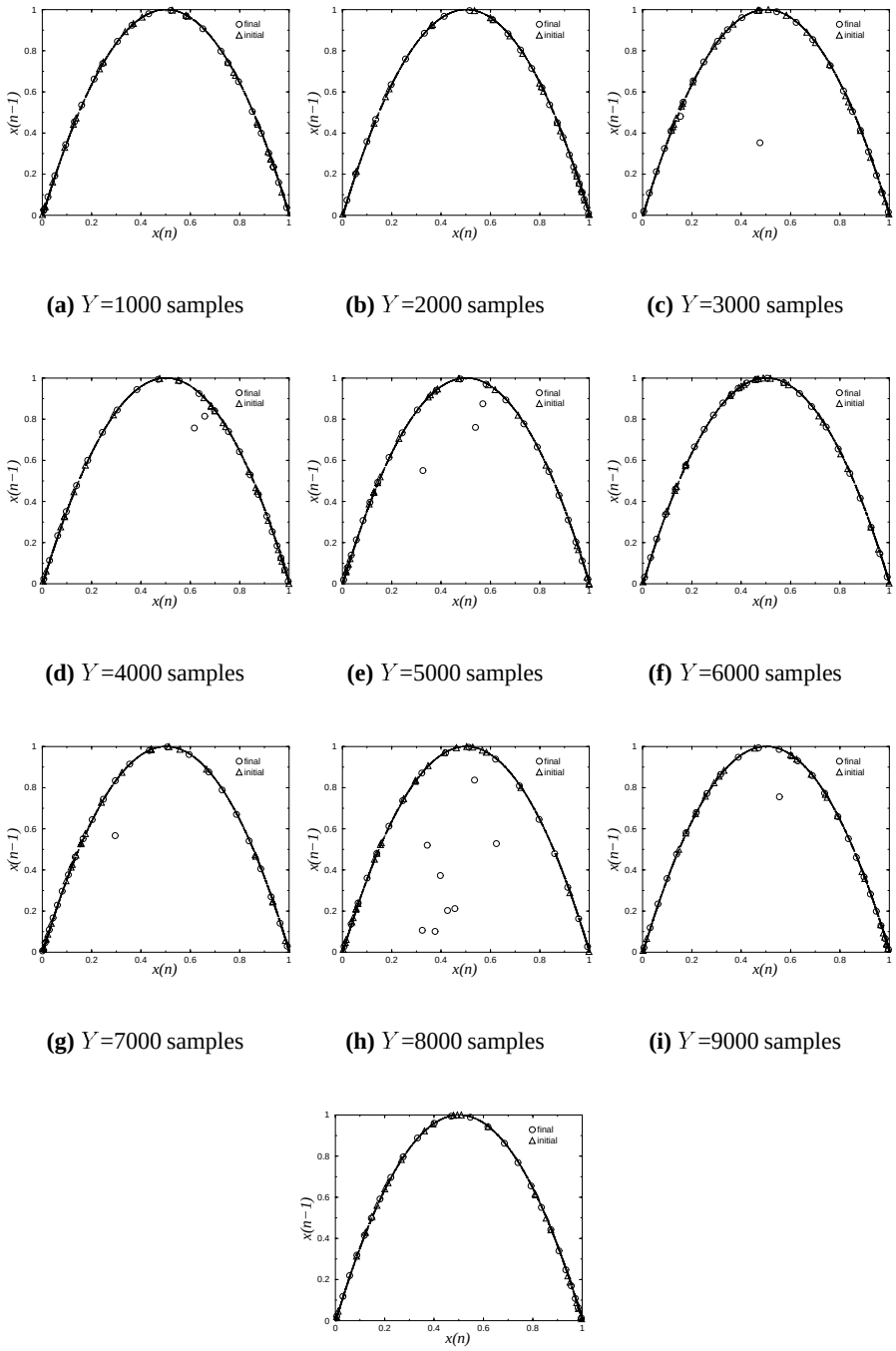


Figure 4.11: *Centre positions in 1-dimensional space for RBFN's using an embedding dimension of 1. These centres correspond to those used by the RBFN's to obtain the prediction results in Figure 4.10, for the cases where an embedding dimension N of 1 was used. The y-axis represents the range of the Logistic map data in 1-dimensional space. On the x-axis, the centres selected using the optimal adaptive k-means algorithm, OAKM method, and the centres selected at random, using the RSC method, are displayed as follows. At x=1,2,3,...,10 the centres selected using the OAKM algorithm are displayed for training lengths of 1000 samples, 2000 samples, 3000 samples,...,10,000 samples, respectively. At x=1.5,2.5,...,10.5, the centres selected at random from the training data are displayed for training lengths of 1000 samples, 2000 samples,...,10,000 samples, respectively. For a given training length the randomly selected centres were used to both initialise the OAKM algorithm and as centres for the RBFN's that used the RSC method. For a given training length and centres selection technique, the same centres were used by both the NRBFN and the UNRBFN.*



(a) Y=1000 samples **(b)** Y=2000 samples **(c)** Y=3000 samples

(d) Y=4000 samples **(e)** Y=5000 samples **(f)** Y=6000 samples

(g) Y=7000 samples **(h)** Y=8000 samples **(i)** Y=9000 samples

(j) Y=10000 samples

Figure 4.12: *Centre positions of RBFN's with embedding dimension of 2, in 2-dimensional space, for training lengths Y of 1000 up to 10,000 samples. Initial centres selected using RSC method, final centres obtained from the initial centres using the OAKM method.*

Training length [samples]	**Width [RSC method]**	**Width [OAKM method]**
1000	0.194027	0.197891
2000	0.199426	0.195599
3000	0.176180	0.198082
4000	0.199232	0.197051
5000	0.197071	0.196827
6000	0.199345	0.195937
7000	0.196257	0.197474
8000	0.199112	0.198145
9000	0.198592	0.197568
10000	0.198552	0.197907

Table 4.1: *Widths for NRBFN's and UNRBFN's for $N=1$, $M=25$ case.*

Training length [samples]	**Width [RSC method]**	**Width [OAKM method]**
1000	0.226046	0.221207
2000	0.225085	0.224215
3000	0.224554	0.223252
4000	0.225090	0.223173
5000	0.225856	0.222547
6000	0.225938	0.220350
7000	0.223248	0.224809
8000	0.225924	0.220792
9000	0.224663	0.222932
10000	0.224458	0.223575

Table 4.2: *Widths for NRBFN's and UNRBFN's for $N=2$, $M=25$ case.*

embedding delay theorem is a sufficient, but not necessary condition for *dynamic reconstruction* [29]. It was found for the Logistic map data, that to achieve the same prediction performance as a RBFNP with an embedding dimension of one, a RBFNP with embedding dimension greater than one had to use many more kernels than the RBFNP with an embedding dimension of one. This will be discussed further below. Additionally, from Figure 4.7 it can be seen that there are no obvious minima, and therefore, the mutual information plot does not yield an obvious choice of embedding delay for a RBFNP, according to the discussion in section 2.2.7. However, through a trial and error procedure it was found that a value of embedding delay equal to 1 sample produced the best NMSE prediction results.

For the cases where N=1, no significant difference in performance between the NRBFNP and the UNRBFNP was observed. However for the N=2 cases, the NRBFNP performed better then the UNRBFNP. The RBFN prediction results were better for the N=1 cases than for the N=2 cases. For both choices of N, 25 kernels were used in each RBFNP. So, from the above observations, it would appear that 25 kernels managed to sufficiently span the 1-D input space, for the N=1 cases, but 25 kernels did not sufficiently span the sparser 2-D input space for the cases where N=2. In the 2-D case, where there were not enough kernels to sufficiently span the input space, the NRBFNP managed to span the input space better than the UNRBFNP. The input space covered by normalised kernels, with respect to the space covered by un-normalised kernels, is discussed in [4, 118, 122]. A more specific explanation for the improvement in prediction performance when normalised kernels are used is that a NRBFN can exhibit both localised *and* non-localised behaviour [121], whereas an UNRBFN only features localised behaviour. Described loosely, a normalised kernel will display localised behaviour if it is surrounded by other kernels, whereas a normalised kernel which is not close to other kernels will exhibit non-localised, sigmoid-like, behaviour [121].

RSC method versus OAKM method

In the cases where N=1, there were no significant differences between the performance of the RBFNP that used the OAKM centres selection method and the performance of the RBFNP that used the RSC centres selection method. For the cases where N=2, it was observed that although both the RSC and the OAKM centres selection techniques resulted in RBFNP's whose prediction performance varied with training length, the RBFNP which used the OAKM method suffered less performance variability than did the RBFNP which used the RSC method. It should be noted that, see Figures 4.11 and 4.12, it took 10,000 training samples before the

input space (for both the N=1 and N=2 cases) was, approximately, equally partitioned (into M clusters) by the OAKM method. From the results in Figure 4.10, the equal partitioning of the input space appears to result in good prediction performance.

The variability of the prediction results for the RBFNP which used the RSC method in the 2-D input space is simply explained by considering the random area in the 2-D space that the RBFN's 25 kernels covered, for each different set of random centres. The 25 kernels did not manage to sufficiently span the input space, and the different sub-space that each set of centres covered, produced different prediction performance results. The variability of the prediction results for the RBFNP which used the OAKM method is understood by looking at Figure 4.12. Figure 4.12 shows that the OAKM method moved the centres around the attractor, and even off the attractor, which had an impact on the prediction performance of the RBFNP. It should be noted however, that despite the OAKM method moving some kernels off the attractor, the OAKM method can still produce better results than the RSC method, if the kernels left on the attractor are better distributed across the attractor than those selected at random using the RSC method (given that the same number of kernels are used in each centres selection technique, as is the case in this example). This is the case, for example, in Figure 4.12(h). In that Figure, the final kernel positions of the OAKM method still managed to cover the attractor better than the initial centres selected using the RSC method, despite some final centres being moved off the attractor. The effect on prediction performance for this example is shown in Figures 4.10(a) and 4.10(c): for the case N=2, and a training length of 8000 samples, the NRBFNP which used the RSC method only had a prediction performance of around -45dB, whereas the NRBFNP which used the OAKM method, had a prediction performance of around -85dB.

This variability in the sparser 2-D space can, however, be eliminated by using enough kernels to sufficiently span the input space. Doing so not only eliminates the variability associated with centres selection, but also improves prediction performance. This is shown in Figure 4.13. For comparison with Figure 4.12, the centres obtained using the RSC method for the NRBFNP and the UNRBFNP, using a RNG seed of 1, are shown in Figure 4.14.

From the above results, the crucial aspect of centres selection seems to lie in the choice of a suitably large number of kernels which can sufficiently span the input space, rather than trying to partition the input space into M equal clusters. However, without prior knowledge of the input space, the OAKM centres selection technique does seem offer an advantage over the RSC method: it results in a RBFNP that suffers less prediction performance variability.

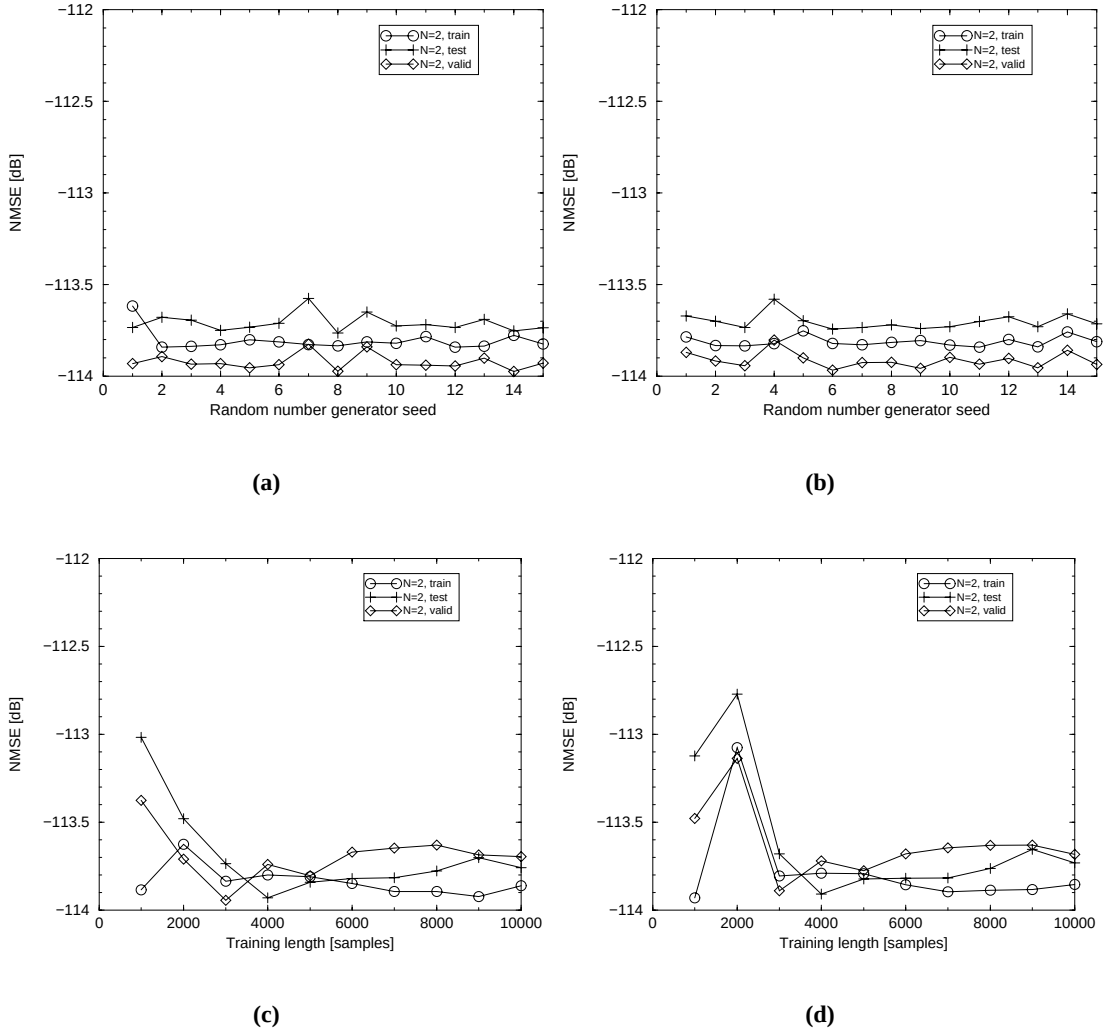


Figure 4.13: *1-step ahead nonlinear prediction of Logistic map data using RBFNP's with an embedding dimension of 2, 100 kernels and an embedding delay of 1 sample. Results are shown for (a) a NRBFNP and (b) an UNRBFNP with centres selected using the RSC method, for different sets of random centres which were obtained by changing the seed to the RNG which was used to pick the centres at random from the training data. The training length for these RBFNP's was 3000 samples. Results are shown for (c) a NRBFNP and (d) an UNRBFNP with centres selected using the OAKM method. The seed used by the RSC initialisation part of the OAKM method was 3.*

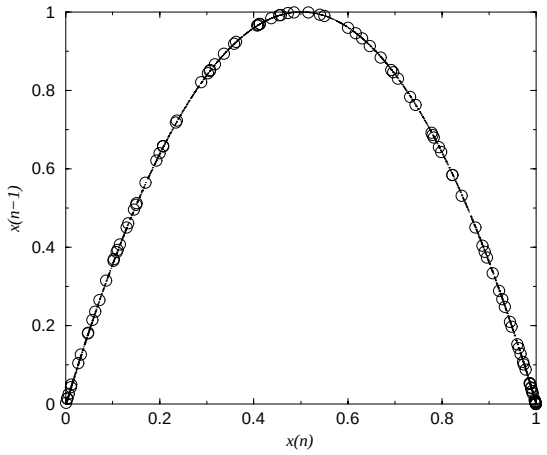


Figure 4.14: *Centre positions in 2-dimensional space for centres obtained using the RSC method, with training length size of 3000 samples, an embedding delay of 1 sample, 100 kernels and a RNG seed of 1. These centres correspond to those used by the RBFN's to obtain the prediction results in Figures 4.13(a) and 4.13(b), for a random number generator seed of 1.*

4.6.4 Nonlinear prediction using a VSFP

Prediction results are shown for a cubic (*i.e.* order=3) VSFP, with an embedding dimension of 10 and an embedding delay of 1 sample, in Figure 4.15. As can be seen, the VSFP achieves similar performance to that of the RBFNP's discussed in section 4.6.3, which used enough kernels to sufficiently span the input space selected.

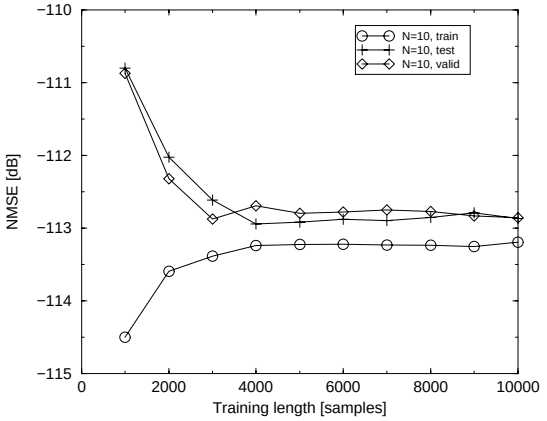


Figure 4.15: *1-step ahead nonlinear prediction of Logistic map data using a cubic VSFP, with an embedding dimension of 10 and an embedding delay of 1 sample.*

4.6.5 Recursive prediction

Figure 4.16(a) shows recursive prediction of the Logistic map data, using a 25 tap LP with a training length of 3000 samples. The LP was seeded initially with the first vector from the testing data set. As would be expected from the results in section 4.6.2, the LP failed to recursively predict the Logistic map data. Figure 4.16(b) shows recursive prediction of the Logistic map, using an UNRBFNP with an embedding dimension of 2, 100 kernels, an embedding delay of 1 sample, and a training length of 3000 samples. The UNRBFNP was seeded initially with the first vector from the testing data set. The UNRBFNP successfully managed to recursively predict the Logistic map time series. The estimate of correlation dimension for the recursively predicted time series was 1.07, which agrees with the estimate for the original time series which was also estimated at 1.07. These values were estimated using 35,000 samples. Note that Takens' embedding delay theorem would suggest that a minimum embedding dimension size of 3 should be used to recursively predict the Logistic map data, however, dynamic reconstruction was achieved using an embedding dimension of only 2.

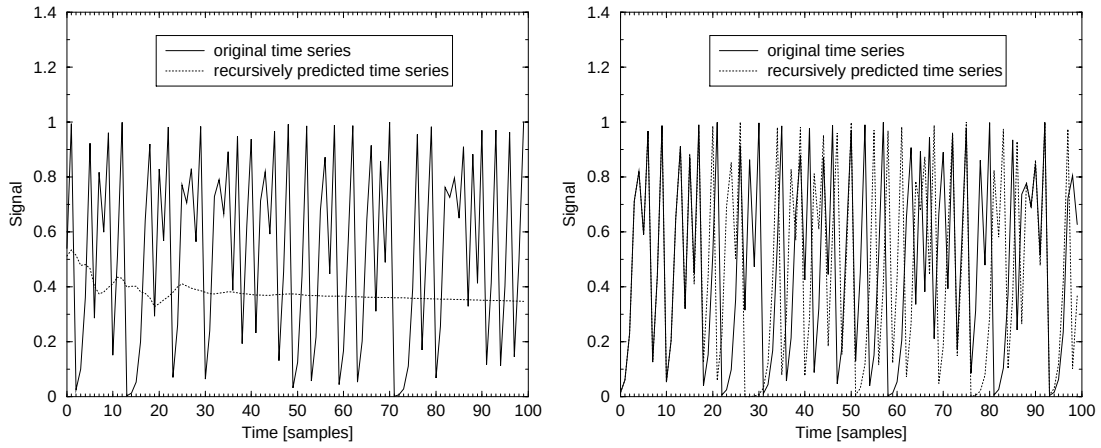


Figure 4.16: *Recursive prediction of Logistic map data using, (a) a LP with 25 taps, and (b) an UNRBFNP with an embedding dimension of 2, an embedding delay of 1 sample, and 100 kernels. The RSC method was used to pick the UNRBFNP's centres, and the seed used by the RNG in the RSC method was 1. 3000 samples were used to train both predictors, and the first input vector from the testing data set was used to seed both predictors, for the recursive prediction task. The prediction horizon (*i.e. *the point where the original and recursively predicted series begin to diverge) can be seen to be approximately 17 samples in Figure 4.16(b).*

4.7 Prediction of Lorenz data

4.7.1 Lorenz attractor

The following coupled system of three nonlinear differential equations [29],

$$\begin{aligned} \frac{dx(t)}{dt} &= -\sigma_L x(t) + \sigma_L y(t) \\ \frac{dy(t)}{dt} &= r_L x(t) - y(t) - x(t)z(t) \\ \frac{dz(t)}{dt} &= -b_L z(t) + x(t)y(t) \end{aligned} \tag{4.20}$$

where σ_L, r_L, and b_L are dimensionless, describe the dynamics of the Lorenz attractor. For σ_L=10, $b_L = \frac{8}{3}$, the system behaves chaotically, whenever the Rayleigh number, r_L, exceeds a critical value, which is approximately 24.74 [31]. Equation (4.20) can be solved for $x(t)$ using a 4^{th} order Runge-Kutta [87] technique with a suitable discrete step-size to produce a discrete Lorenz time series. For the work in this thesis, a step-size of 0.05, and a value of r_L=28 were used to obtain Lorenz time series data. Figure 4.17 is a plot of the Lorenz time series in 3-D state space, or, in other words it is a plot of the Lorenz attractor.

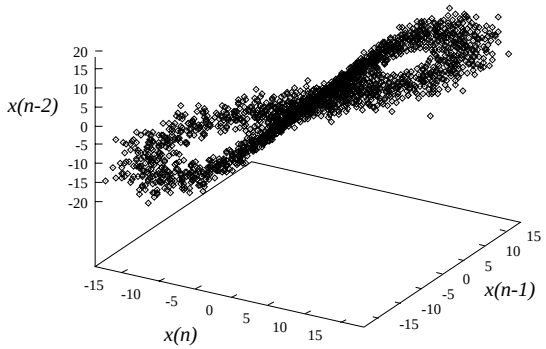


Figure 4.17: *Plot of the Lorenz attractor using 2000 data points.*

The correlation dimension estimate for the Lorenz data was evaluated using the method of maximum likelihood, which was discussed in Chapter 2. The estimate for the maximum likelihood correlation dimension D_{ML} was 2.11. A plot of the mutual information of the Lorenz data is given in Figure 4.18.

1-step ahead prediction of the Lorenz data was carried out using linear, RBFN and VSF pre-

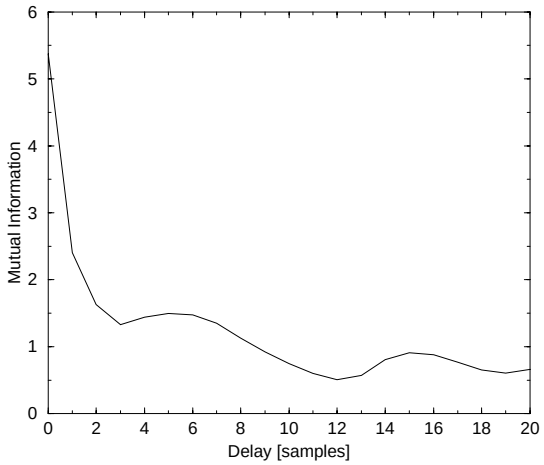


Figure 4.18: *Plot of the mutual information for the Lorenz data.*

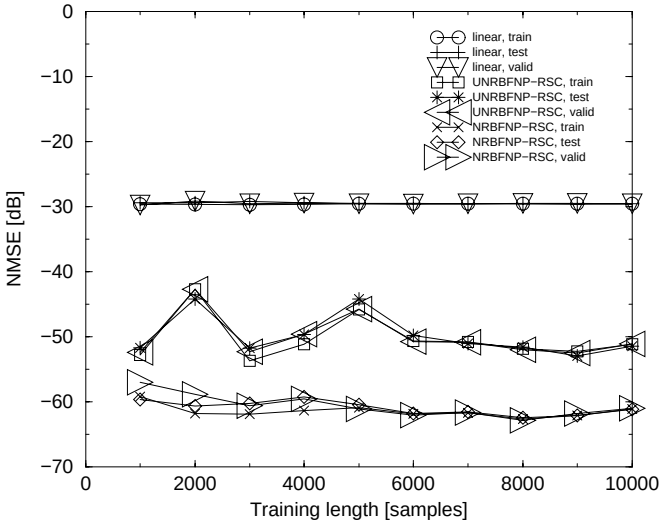
dictors. This prediction analysis is now discussed.

4.7.2 Prediction of Lorenz data

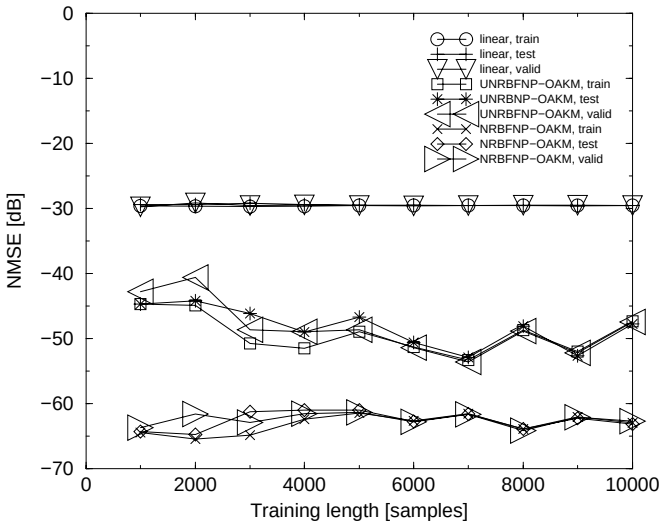
The 1-step ahead prediction performance of linear, VSF and RBFN predictors is reported in this section. Previous work on the prediction of Lorenz data is reported in [99, 100].

Linear and RBFN prediction results are presented in Figure 4.19. From Takens' embedding delay theorem a suitable choice for embedding dimension, for the prediction of the Lorenz signal, would be 7 or greater. From the plot of mutual information in Figure 4.18, a suitable choice of embedding delay would be 3 samples. However, from trial and error experiments, the optimal values for a RBFN's embedding dimension and embedding delay, in terms of the NMSE performance measure, were 3 and 1 sample, respectively. Results are presented in Figure 4.19 for RBFN's with an embedding dimension of 3, and embedding delay of 1 sample, trained using the RSC method (Figure 4.19(a)) and the OAKM method (Figure 4.19(b)) to select centres. Results, for comparison, are also shown for a LP with 30 taps.

From the results in Figure 4.19, 100 kernels seems sufficient to adequately span the input 3-D space, and therefore (as discussed in section 4.6.3), there was found to be no significant difference in terms of NMSE between a RBFNP which used the RSC method, and one which used the OAKM method. The nonlinear RBFNP's were able to exploit the nonlinear nature of the Lorenz signal, and therefore performed better than the LP. As there was found to be no



(a)



(b)

Figure 4.19: *1-step ahead nonlinear prediction of Lorenz data using RBFNP's with an embedding dimension of 3, 100 kernels and an embedding delay of 1 sample. Results are shown for (a) a RBFNP with centres selected using the RSC method, and for (b) a RBFNP with centres selected using the OAKM method. Results are also shown for a 30 tap LP.*

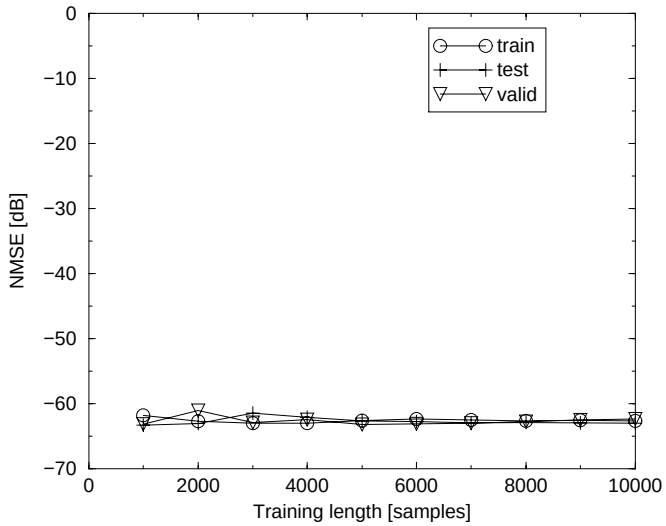


Figure 4.20: *1-step ahead nonlinear prediction of Lorenz data using a cubic VSFP, with an embedding dimension of 4 and an embedding delay of 1 sample.*

significant difference in NMSE between using the two different centres selection techniques, for both the Logistic map data and the Lorenz data, the computationally more efficient RSC method was preferred to the OAKM method, for the prediction analysis reported in the remainder of this chapter. For convenience, a RBFNP with centres selected using the RSC method will be referred to as a RBFNP-RSC, and a RBFNP with centres selected using the OAKM method will be referred to as a RBFNP-OAKM, in the remainder of this thesis.

Results for a cubic VSFP with an embedding dimension of 4 are shown in Figure 4.20. As was the case for a RBFNP, the VSFP was able to exploit the nonlinear nature of the Lorenz signal.

4.7.3 Recursive prediction of noisy Lorenz data

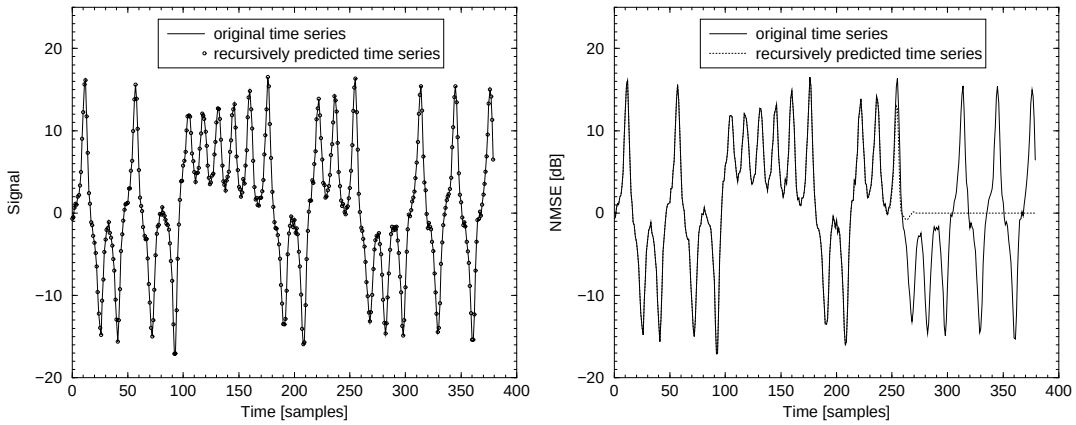
In [29] Haykin and Principe presented results for the recursive prediction of Lorenz data in white Gaussian noise, with a SNR of 25dB. They presented results for an UNRBFN which showed that it could not recursively predict the noisy Lorenz data. They also showed results for a RUNRBFN, which was able to recursively predict the noisy Lorenz data. Haykin and Principe used a fully regularised solution.

To assess the usefulness of regularisation as a tool for being able to reconstruct the dynamics of a chaotic signal, Lorenz data was generated as discussed in section 4.7.1, and white Gaussian

noise was added to the Lorenz data. The SNR was chosen to be 25dB.

A fully regularised UNRBFN was implemented with the same number of kernels as Haykin and Principe used: 400. An embedding dimension of 20, and an embedding delay of 1 sample were used, just as Haykin and Principe had used. The training data file used was 420 samples long, which provided 400 input vectors of length 20, and 400 1-step ahead samples: the 400 input vectors were used as centre positions for the 400 kernels. The testing data file consisted of the next again 420 samples from the noisy Lorenz file, and the validation set, consisted of the next again 420 samples. Haykin and Principe did not report the particular value of the regularisation parameter λ which resulted in successful recursive prediction, but they did say that the value of λ used varied between 10^{-14} and 10^{-2}, depending on the specific data set analysed. Recursive prediction was attempted using the fully regularised UNRBFNP-RSC described above for regularisation parameter values: 10^{-14}, 10^{-13}, 10^{-12},..., 10^{-2}. However, unlike the results reported by Haykin and Principe, for the range of regularisation parameter values discussed above, regularisation did not result in recursive prediction working. It should be noted that the fully regularised solution implemented here is different to that used by Haykin and Principe in that Haykin and Principe used a weighted norm [24] distance measure, as opposed to the Euclidean distance measure used in this work. The fully regularised approach resulted in over-fitting to the training data for regularisation parameter values of $10^{-14} \rightarrow 10^{-5}$. For this range of λ, the recursive prediction worked for the training data, in actual fact the recursively predicted signal was identical to the original time series, see Figure 4.21(a). For the range of $\lambda = 10^{-4} \rightarrow 10^{-2}$, recursive prediction failed for the training data, see Figure 4.21(b). For $\lambda = 10^{-14} \rightarrow 10^{-2}$ recursive prediction failed for the testing data. The initial input vector from the training data set was used as the seed for the recursive prediction of the training data, and the initial input vector from the testing data set was used as the seed for the recursive prediction of the testing data set. As discussed in [17], in order to claim that the underlying dynamics of a chaotic process have been captured successfully, recursive prediction must be shown to work on non-training data. Therefore, it can be stated that the fully regularised UNRBFNP-RSC was not able to successfully recursively predict the noisy Lorenz time series.

Recursive prediction results are presented in Figure 4.22 for a LP, an UNRBFNP-RSC and a NRBFNP-RSC. As can be seen, the LP and UNRBFNP-RSC both failed to capture the underlying dynamics of the Lorenz data. However, the NRBFNP-RSC did manage to capture the underlying dynamics.



(a) **(b)**

Figure 4.21: *Recursive prediction of noisy Lorenz data (SNR=25dB) using a fully regularised UNRBFNP-RSC with an embedding dimension of 20, 400 kernels, an embedding delay of 1 sample and (a) a regularisation parameter of* 10^{-5}*, (b) a regularisation parameter of* 10^{-4}*. Results are shown for the training data sets.*

Recursive prediction was also attempted using approximate regularisation (see section 4.4.5). For an UNRBFNP-RSC with an embedding dimension of 20, an embedding delay of 1 sample, 400 kernels, and a training length of 2000 samples, approximate regularisation was used to train the output layer weights, for a range of regularisation parameters: $\lambda = 10^{-1}, 10^{-2}, ..., 10^{-14}$. Recursive prediction failed for each regularisation parameter value. Note that in order to compare the effect of the regularisation parameter on the approximately regularised solution, with that for the fully regularised solution, the effect of training length must be taken into account. This is best explained by considering the cost function [29] for the regularisation technique which is given in equation (4.21),

$$\mathrm{err} = \arg\min_{\mathrm{err}\in S}\left\{\sum_{n=N}^{Y-1}(x(n+1)-\hat{x}(n+1))^2+\lambda||\mathbf{P}\mathrm{err}||^2\right\} \tag{4.21}$$

where $x(n+1)$ is the next sample, $\hat{x}(n+1)$ is the predictor's estimate of the next sample, Y is the length of the training data file, $\lambda||\mathbf{P}\mathrm{err}||^2$ is known as the regularising term which represents a model complexity-penalty option. S is the input space of dimension N. Obviously, as Y is increased the errors will increase. To normalise the cost function with respect to training length, the regularisation parameter can be modified as follows. Given a regularisa-

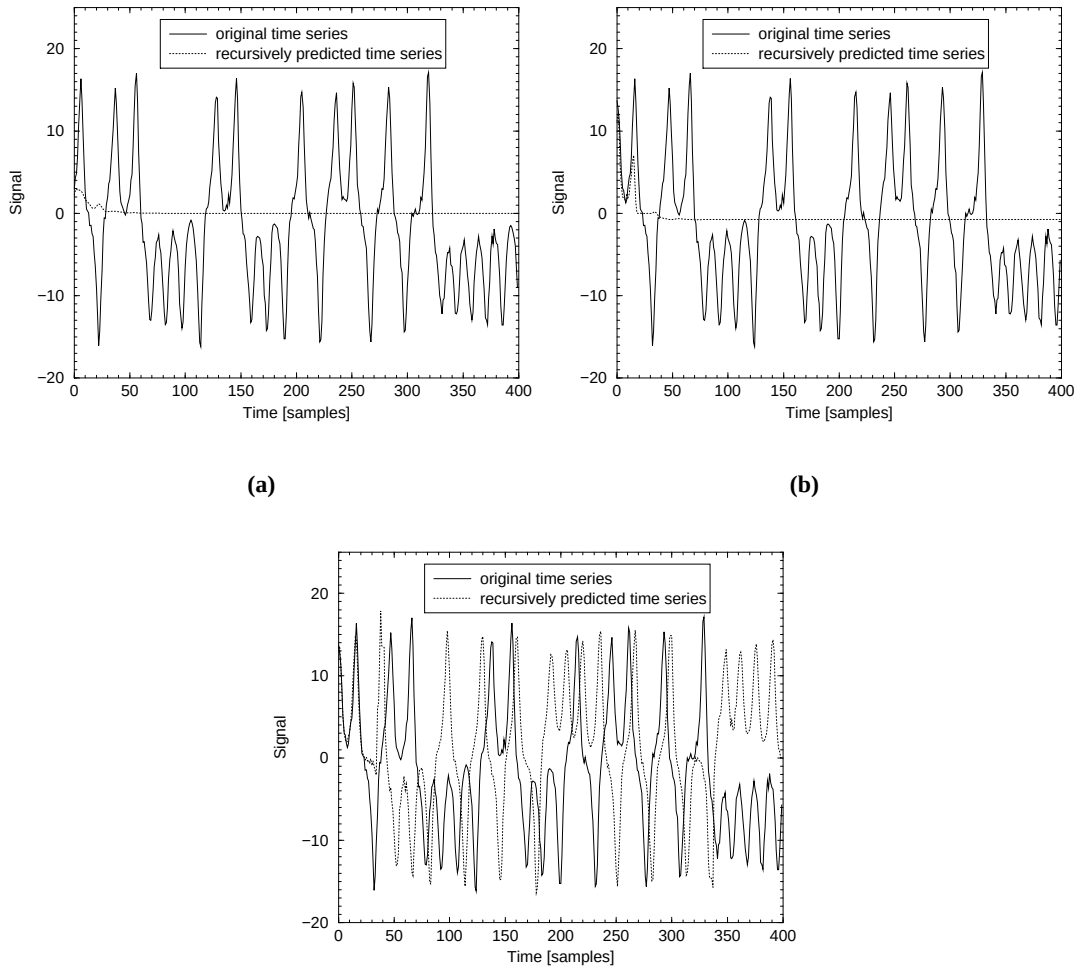


Figure 4.22: *Recursive prediction of noisy Lorenz data (SNR=25dB) using (a) a 30 tap LP, (b) an UNRBFNP-RSC, and (c) a NRBFNP-RSC. The RBFNP's both used an embedding dimension of 20, 400 kernels and an embedding delay of 1 sample. In all cases recursive prediction was initialised with the first vector from the testing data set.*

tion parameter range for training length Y_1, the required regularisation parameter range for a training length of Y_2 can be obtained by multiplying the range for Y_1 by $\frac{Y_2}{Y_1}$. In this way, the effect of regularisation parameter can be compared for the two different training lengths. For the approximate regularisation recursive prediction mentioned above, a range of regularisation parameters, $\lambda = 10^{-1}, 10^{-2}, ..., 10^{-14}$, was used. The normalisation method just described, was used to obtain this range, so that the effect of regularisation parameter used for the fully regularised UNRBFN results could be compared with the results for the approximately regularised solution. The multiplication factor is actually $\frac{2000}{420}$, which is 4.76. This would mean the appropriate range of the regularisation parameter for the approximately regularised solution should be $\lambda = 4.76 \times 10^{-2} \rightarrow 4.76 \times 10^{-14}$. The range used, $\lambda = 10^{-1}, 10^{-2}, ..., 10^{-14}$, incorporates these values.

4.7.4 Further recursive prediction analysis using a NRBFNP-RSC

Having established that a NRBFNP-RSC can be used to recursively predict noisy Lorenz data, it is useful to reconsider the size of the embedding vector used for this recursive prediction exercise: the principal question to ask is, is such a large embedding dimension really necessary? Haykin and Principe suggested using an embedding dimension that was obtained as follows. Having established a suitable embedding dimension N using a correlation dimension estimate, and a suitable embedding delay τ using the mutual information of the data, Haykin and Principe then suggested using an embedding dimension of $N\tau$, with an embedding delay of 1 sample.

In the noise-free case, it has been found that Lorenz data can be recursively predicted using a NRBFNP-RSC with an embedding dimension of 3 (less than Takens' embedding delay theorem criteria would suggest), 100 kernels, and an embedding delay of 1 sample. This is illustrated in Figure 4.23(a). However, in the case with noise added, and a SNR of 25dB, using an embedding dimension of 3, 400 kernels, and an embedding delay of 1 sample, a NRBFNP-RSC failed to recursively predict the noisy Lorenz data. The maximum likelihood correlation dimension estimate for the noisy Lorenz data was estimated using 35,000 samples to be 2.38. This gives a minimum embedding dimension of 7, for dynamic reconstruction using Takens' embedding delay theorem. The mutual information plot of the noisy Lorenz data revealed that a suitable choice for embedding delay would be 3 samples. Recursive prediction for a NRBFNP-RSC with an embedding dimension of 7 and an embedding delay of 3 samples is shown in Figure 4.23(b). As can be seen, the NRBFNP-RSC managed to not only successfully recursively

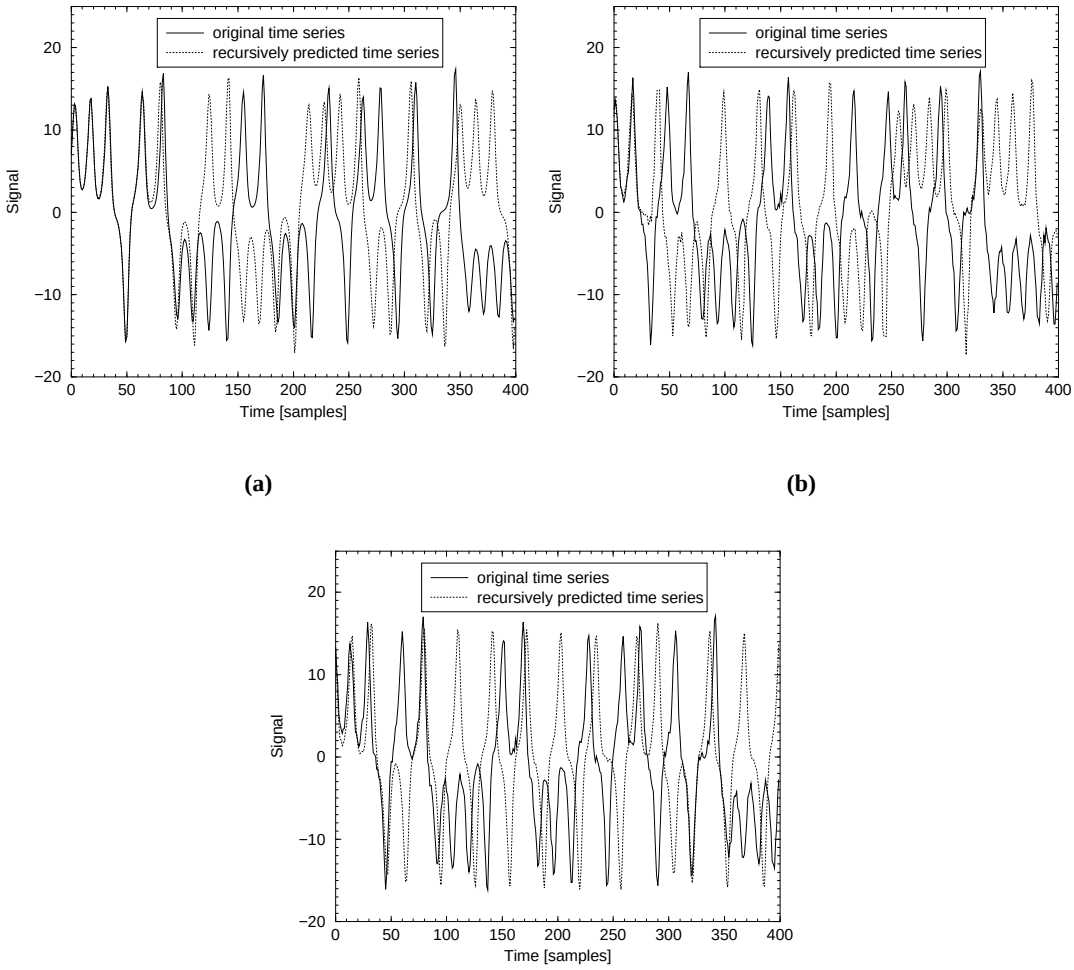


Figure 4.23: *Recursive prediction using a NRBFNP-RSC with a training length of 2000 samples and (a) an embedding dimension of 3, 100 kernels and an embedding delay of 1 sample on noise-free Lorenz data, (b) an embedding dimension of 7, 400 kernels and an embedding delay of 3 samples on noisy Lorenz data (SNR=25dB), (c) an embedding dimension of 7, 400 kernels and an embedding delay of 1 sample on noisy Lorenz data (SNR=25dB). In each case recursive prediction was initialised using the first vector from the testing data set.*

predict the noisy Lorenz data, but it also seemingly managed to capture as much signal detail as did the NRBFNP-RSC whose results were reported in Figure 4.22(c). To explain what is meant by "signal detail", consider the recursive prediction results for the NRBFNP-RSC with an embedding dimension of 7, and an embedding delay of 1 sample, shown in Figure 4.23(c). In Figure 4.23(c) it can be seen that although recursive prediction did not completely fail, it failed to capture much of the original signal's detail: when the original time series and the recursively predicted time series diverged, the recursively predicted time series merely seemed to oscillate between a positive and negative signal amplitude. After the recursively predicted time series diverged from the original time series in Figure 4.23(b), 3-peak and 4-peak features, observable in the original time series, were also observable in the recursively predicted time series. Effectively the NRBFNP-RSC with an embedding dimension of 7 and an embedding delay of 3 samples managed to capture the underlying dynamics of the signal, whereas the NRBFNP-RSC with an embedding dimension of 7 and embedding delay of 1 sample did not. This can be explained by considering the evolution of the NRBFNP-RSC input vectors in state space. Increasing the embedding delay has the effect of "opening out" the attractor in state space, which reduces the likelihood that noise will cause any vector to erroneously evolve (or jump) to the wrong part of the attractor. Avoiding such erroneous evolution eventualities results in correctly capturing the underlying dynamics of the chaotic signal in question.

Table 4.3 shows the maximum likelihood correlation dimension estimates, along with training, testing, and validation NMSE's for the following NRBFNP-RSC's (which were discussed above): the NRBFNP-RSC with an embedding dimension of 7, and an embedding delay of 1 sample, the NRBFNP-RSC with an embedding dimension of 7, and an embedding delay of 3 samples, and for the NRBFNP-RSC with an embedding dimension of 20, and embedding delay of 1 sample. As already discussed, each predictor was trained using a training length of 2000 samples, and each predictor was used to recursively produce a time series of length 35,000 samples, using the first vector of the testing data set to initialise the recursive prediction process.

As can be seen from Table 4.3, the recursively produced time series with the closest correlation dimension estimate, to that of the original time series is that produced by the NRBFNP-RSC with an embedding dimension of 7 and an embedding delay of 3 samples. A fact that implies this NRBFNP-RSC is the one which has managed to produce a time series with the closest (out of the three networks listed in Table 4.3) long term properties (see section 4.5.2) to those of the

NRBFNP-RSC embedding dimension	**NRBFNP-RSC embedding delay**	**Correlation dimension estimate (D_{ML})**	**Training NMSE [dB]**	**Testing NMSE [dB]**	**Validation NMSE [dB]**
7	1	2.03	-23.63	-20.97	-20.85
7	3	2.28	-21.59	-19.16	-17.92
20	1	2.25	-21.43	-19.64	-18.78

Table 4.3: *Recursive prediction of noisy Lorenz data (SNR=25dB), using a NRBFNP-RSC.*

original time series. A reason why the D_{ML} estimate was slightly poorer for the NRBFNP-RSC with an embedding dimension of 20, than for the NRBFNP-RSC with embedding dimension of 7 and embedding delay of 3 samples, is that this embedding dimension is actually too large, and as a result the additive Gaussian noise has degraded the quality of the dynamic reconstruction [29]. It should also be noted that the network that best managed to capture the underlying dynamics was not the network with the best NMSE performance values. The best predictor network in terms of NMSE was the NRBFNP-RSC with an embedding dimension of 7 and an embedding delay of 1 sample. The correlation dimension estimate for the time series recursively produced by this network was 2.03, the poorest estimate of the three estimates in Table 4.3, supporting the comments made above about the level of detail captured in the recursively predicted time series produced by the NRBFNP-RSC with an embedding dimension of 7 and an embedding delay of 1 sample.

So to answer the question posed above, a large embedding dimension of 20 *can* be used to recursively predict noisy Lorenz data, and thus to capture the underlying dynamics, however, similar (and slightly better) results can be achieved using a substantially smaller embedding dimension, with a suitable embedding delay.

4.8 Summary

In this chapter the structure and training of two nonlinear models, the Volterra series filter and the radial basis function network, were discussed. In particular, the network architecture and the selection of the kernel centres for a RBFNP were considered in some detail, using two well known chaotic signals: Logistic map and Lorenz data. It was shown that a RBFNP with normalised Gaussian kernel functions could outperform a RBFNP that used un-normalised Gaussian kernel functions, in terms of NMSE. For a given RBFNP, little NMSE performance

difference was observed for the two different kernel centres selection techniques considered: the RSC method and the OAKM method. The RSC method was therefore preferred for the prediction of these chaotic signals as it is computationally less expensive. It was found that the embedding delay which resulted in the best NMSE prediction estimate for a RBFNP was 1 sample, for both Logistic map and Lorenz data. It was also shown that a NRBFNP-RSC could capture the underlying dynamics of a chaotic signal corrupted by noise. Attempts using an UNRBFNP-RSC and a UNRBFNP-RSC with regularisation failed to successfully recursively predict the same data.

Chapter 5
Nonlinear prediction of sea clutter

5.1 Introduction

Attention is now turned to the task of predicting sea clutter. As discussed in section 2.2.11, research has been carried out [9, 12–18] which has shown that sea clutter is predictable using a nonlinear predictor (NLP). As already mentioned, these prediction studies were part of a broader agenda: to prove that sea clutter is a chaotic process. Leung and Lo [12] demonstrated that a global NLP could achieve better prediction performance on sea clutter amplitude data sets than a global linear predictor (LP), in terms of mean square error (MSE). The interest in the predictability of sea clutter, for maritime surveillance radar, is in the possible incorporation of a predictor into a detection algorithm. Indeed, Leung and Lo [12] showed how a NLP could be used as the basis of a detection technique: a nonlinear predictor-detector (NLPD). This NLPD technique performed better than both a linear predictor-detector (LPD), and a *"standard"* non-coherent detector (although the details of this standard detector were not specified). The topic of predictor-detectors for the clutter data sets described in Appendix A is considered in Chapter 7. In [9] Haykin and Leung carried out recursive prediction [29] (see section 4.5.2) on one of the sea clutter data sets used by Leung and Lo [12] in their work. A NLP was found to be able to recursively predict the clutter, however a LP was not able to do this. These prediction results were taken as evidence that sea clutter is nonlinearly predictable. The use of a LP comparison with the NLP in the above mentioned papers seems both sensible and necessary: the data might only have been linearly predictable, in which case a NLP would simply be trying to approximate a linear predictor function, and so a simpler LP would be preferred. Using only a NLP, it is not possible to distinguish between linear predictability and nonlinear predictability. However, in [13–15, 17] many more sea clutter data sets than those analysed in [9, 12] were analysed using NLP's, but there were no reported LP comparisons!

It is the task of this chapter to further investigate the predictability of sea clutter, using the new sea clutter data sets described in Appendix A, and the nonlinear prediction techniques introduced in Chapter 4. It should be noted that no assumptions were made about the sea clutter

data sets (*i.e.* about whether or not they were best modelled as a stochastic or deterministic process) analysed in this thesis. A pragmatic approach was adopted: regardless of sea clutter model, it was investigated whether the clutter was predictable at all, and if so, whether a global NLP could give better prediction performance than a global LP. An investigation of this nature is essential in determining whether or not a NLP can be used to enhance the performance of existing radar detectors (as is claimed in [12, 17]), which is the subject of Chapter 7. Unlike previous work [9, 12–18] which investigated the predictability of sea clutter, prediction analysis was carried out on stochastic surrogate sea clutter data sets. These surrogates were generated using the popular compound K-distribution model [22] (see Chapter 3). Also, recursive prediction was carried out on the real sea clutter data sets described in Appendix A. An investigation into the recursive predictability of sea clutter is particularly relevant bearing in mind the successful sea clutter recursive prediction results presented in [9, 13, 17], and also, bearing in mind the claim made that sea clutter is chaotic [11, 13, 14, 17–21]. Moreover, in [17] recursive prediction was used by Haykin and Li to demonstrate the good generalisation properties of a nonlinear predictor for sea clutter, after the training phase had been completed.

This chapter is structured as follows. In section 5.2 prediction results for the $12ms^{-1}$ wavetank[1] range gate sea clutter data sets are reported. In section 5.3 the prediction results for the $4ms^{-1}$, $5ms^{-1}$, $6ms^{-1}$, $7ms^{-1}$, $8ms^{-1}$, $9ms^{-1}$, $10ms^{-1}$, and $11ms^{-1}$ wavetank sea clutter data sets are presented. A summary of the prediction results for the wavetank data sets is given in section 5.4, together with an explanation for the source of the predictability of these data sets. In section 5.5 the prediction results for the Dawber clutter data sets are presented. A summary of the prediction results for the Dawber data sets is given in section 5.6. A chapter summary is given in section 5.7.

5.2 Prediction of the $12ms^{-1}$ wavetank range gate data sets

5.2.1 Correlation dimension and mutual information

This section describes the process for choosing the embedding dimension and embedding delay for a RBFNP, for the initial prediction analysis of the $12ms^{-1}$ wavetank sea clutter data sets.

As discussed in Chapter 2, the correlation dimension estimate and mutual information plot for

[1] It should be noted that the convention for referring to the wavetank range gate data sets is that a data set collected during a windspeed of Xms^{-1} will be referred to as the Xms^{-1} wavetank data set.

a data set can be used to embed a chaotic signal. However, as mentioned in sections 4.6 and 4.7, this embedding criterion does not always lead to the best prediction performance, in terms of NMSE.

For the initial prediction analysis of the $12ms^{-1}$ data sets, a pragmatic approach for the selection of a suitable RBFNP embedding dimension and embedding delay was adopted, which is now discussed. The maximum likelihood method (see Chapter 2) was used to estimate the correlation dimension for each $12ms^{-1}$ range gate data set. The D_{ML} estimate for each wind-speed range gate data set, can be found in Table E.1, in Appendix E. Mutual information plots were produced for all 32 range gate data sets.

The D_{ML} estimates for the different $12ms^{-1}$ range gate data sets, were observed to vary between 3 and 8. For a D_{ML} of 3, the minimum embedding dimension N according to Takens' theorem, for dynamic reconstruction, is $N \geq 7$, and for a D_{ML} of 8 the criterion for embedding dimension is $N \geq 17$. However, as observed in sections 4.6 and 4.7, this criterion for dynamic reconstruction does not always result in the best NMSE performance. Indeed, values smaller than this criterion for both the Logistic map and the Lorenz data sets, were found to give better NMSE performance (given the same number of Gaussian kernels for the smaller values of embedding dimension, as for the larger values of embedding dimension). Therefore, for the initial prediction analysis of the wavetank data, a number of different embedding dimensions were chosen: $N = 5, 10, 20, 40$. In other words, some embedding dimensions which did not obey the criterion for dynamic reconstruction, and some that did, were used.

For the choice of a suitable embedding delay for a RBFNP, the mutual information plots for each range gate data set were examined. Unfortunately, most of these plots were found to be monotonically decreasing, and therefore had no minima, and so, according to the criterion in section 2.2.7, a suitable choice for embedding delay from the mutual information plots was not possible, for most range gates. Therefore, a value of 1 sample for the embedding delay, for each range gate was selected, as this had given the best NMSE results in sections 4.6 and 4.7. Additionally, it should be noted that using a large embedding dimension with an embedding delay of 1 sample, is consistent with the approach adopted towards prediction in [17, 29].

An example of a monotonically decreasing mutual information plot for the $12ms^{-1}$ clutter data is given in Figure 5.1, below.

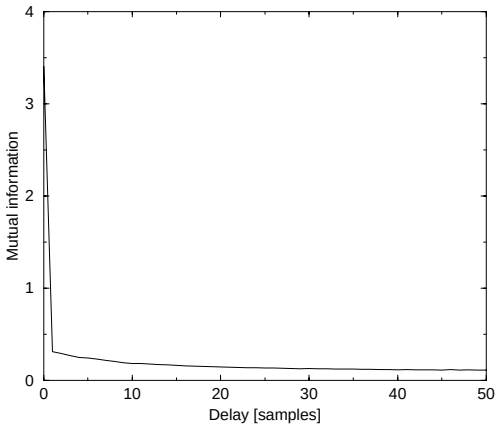


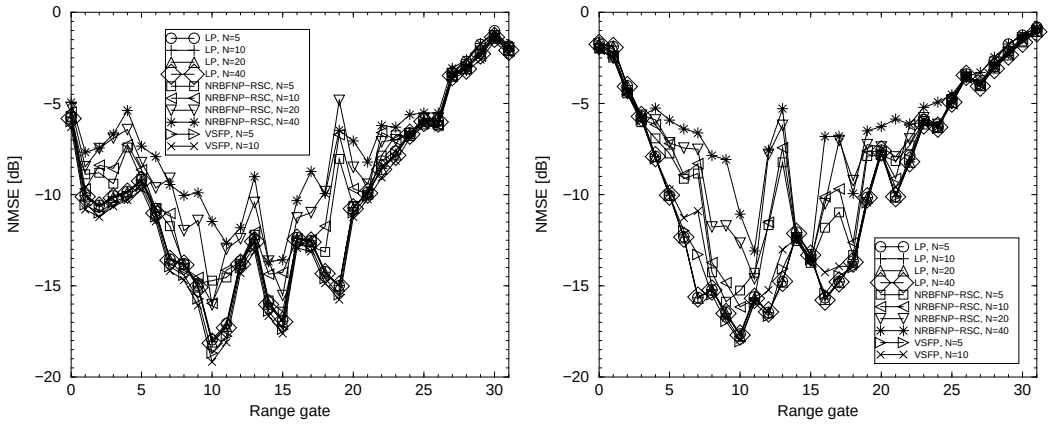
Figure 5.1: *Mutual information for $12ms^{-1}$ range gate 0 data set.*

5.2.2 Initial prediction results

Linear prediction results for 5, 10, 20 and 40 tap LP's are given in Appendix F. Nonlinear prediction results are presented for cubic VSFP's, with embedding dimensions of 5 and 10 and an embedding delay of 1 sample for each case of embedding dimension, in Appendix F. Embedding dimensions of 20 and 40 were not considered on the grounds of computational complexity: a cubic VSFP with an embedding dimension of 20 would have 1770 Volterra kernels!

The choices for embedding dimension and delay, discussed in section 5.2.1, were used for a NRBFNP-RSC. Additionally, a NRBFNP-OAKM, an UNRBFNP-RSC, and an UNRBFNP-OAKM were used with an embedding dimension of 20 and an embedding delay of 1 sample. Each RBFNP used 100 Gaussian kernels. The RBFNP results are given in Appendix F.

Several observations were made from these initial prediction simulations. The most important observation is that for any given embedding dimension, training length, and range gate, a nonlinear predictor was found to be only as good as, or worse than a LP with a number of taps equivalent to the embedding dimension of the NLP, and an equivalent training length. Furthermore, a predictor with 5 taps, or an embedding dimension of 5, had as good performance, or better than the same type of predictor with larger choices for number of taps, or embedding dimension. This was true for the data in each range gate. These two observations are illustrated in Figure 5.2, for all the $12ms^{-1}$ range gate data sets. The apparent linear predictability of sea clutter is further investigated in section 5.4 Note that in Figure 5.2 there is a noticeable variation



(a) **(b)**

Figure 5.2: *Initial 1-step ahead prediction results for the wavetank $12ms^{-1}$ range gate data sets: (a) training and (b) testing data set results. Results are shown for LP's with number of taps N equal to 5, 10, 20, 40, and for NRBFNP-RSC's with embedding dimensions N of 5, 10, 20, 40, with 100 kernels and an embedding delay of 1 sample used in each case, and for cubic VSFP's, with embedding dimensions N of 5 and 10, and an embedding delay of 1 sample in each case. All predictor results are shown for a training length of 10,000 samples.*

in predictability across range gates. This is discussed in more detail in section 5.4.

It was also observed that the NRBFNP's, in general, performed better than the UNRBFNP's, and that there was little difference in terms of NMSE, between the RBFNP's that used the RSC technique to select centres, with those that used the OAKM method. These facts are illustrated in Figure 5.3. This evidence was used to justify the use of RBFNP-RSC's instead of RBFNP-OAKM's in the remainder of this thesis.

Close inspection of the NMSE results for the NRBFNP's in Appendix F reveals that some entries are not numbers, but they are listed as "∞". The reason for obtaining these infinite values will be explained in more detail in the next section.

5.2.3 NRBFNP results: further analysis

As already mentioned, infinite NMSE values have been obtained from some of the NRBFNP simulations listed in Appendix F. For example, the testing data set NMSE results for the NRBFNP-RSC with an embedding dimension of 5 show an occurrence of such a value.

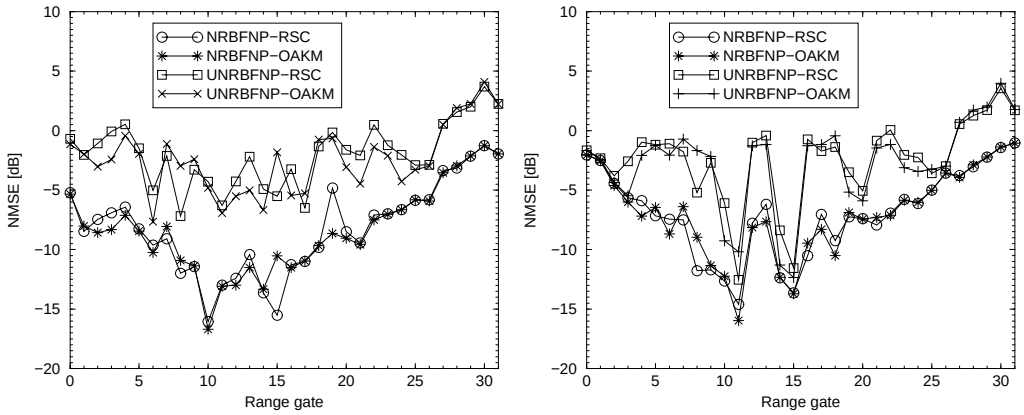


Figure 5.3: *Initial 1-step ahead RBFN prediction results for the wavetank $12ms^{-1}$ range gate data sets: (a) training and (b) testing data set results. Each RBFNP used an embedding dimension of 20, 100 kernels, an embedding delay of 1 sample, and a training length of 10,000 samples.*

The reason for obtaining infinite values is best explained by looking at a case where an infinite NMSE value was encountered. Arbitrarily, range gate 15 is considered for this explanation. A plot of the time series in range gate 15 is given in Figure 5.4.

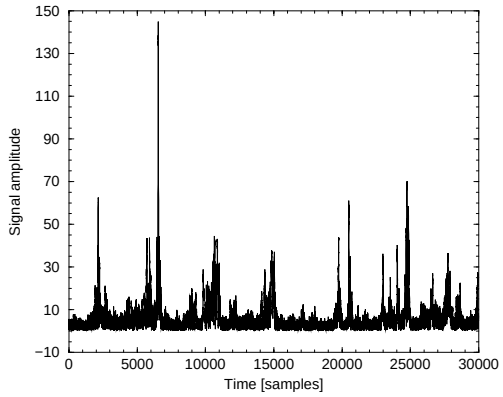


Figure 5.4: *Time series plot of the $12ms^{-1}$ range gate 15 data set.*

From Figure 5.4, it can be seen that the data is very impulsive or "spiky", in that it contains several samples with a very large amplitude (large, relative to the majority of the clutter samples), which are commonly referred to as spikes in the radar literature. These spikes are the cause of the infinite NMSE values: if a NRBFNP input vector of sea clutter, in N-dimensional space, is too far away from all the centres of the network's kernels, then all the kernel outputs will be

zero, and in this case the normalised kernel output becomes infinite. Encountering an infinite NMSE will be referred to as NRBFNP failure from this point onwards. This problem is now examined in greater detail for the case of the NRBFNP-RSC with an embedding dimension of 20, which is given in Appendix F. Specifically consider the testing NMSE's for training lengths of 2000 and 6000 samples, and the validation NMSE's for training lengths of 1000 and 2000 samples.

From these examples, first consider the failure of the NRBFNP-RSC (*i.e.* the occurrence of an infinite NMSE value), on the validation data set for a training length of 1000 samples: where (as discussed in section 4.5.1) the first 1000 samples of range gate 15 were used as the training data set, the second 1000 samples were used as the testing data set and the third 1000 samples were used as the validation data set. From Figure 5.4 it can be seen that in the validation data set for this example (between 2000 and 3000 samples) there are spikes that are much larger than any of the samples in the training data set. Upon closer inspection of the NRBFNP-RSC used, it was observed that the spikes in the validation data set (just after 2000 samples) were large enough, and far enough away from the centres selected at random in the training data, to result in all the kernel outputs being zero[2], and thus an infinite NMSE value was encountered. In this example there were no spikes in the testing data set, and therefore there was no problem with the NMSE value for the testing data. The above explanation can also be applied to the occurrences of the infinite NMSE value, for the testing and validation sets with a training length of 2000 samples, and for the validation set, with a training length of 6000 samples. Rather surprisingly, for the case where the training length is 3000 samples, the largest spike (in between 6000 and 7000 samples) did not result in an infinite NMSE value for the validation data set. After closer inspection of the NRBFNP-RSC, it was found that by chance the RSC method had picked centres from the training data that were close enough to this large spike value to prevent all the kernels from going to zero when this spike was read in, and thus the occurrence of an infinite NMSE value was avoided: the training data set had fortunately contained large enough spikes which allowed such centres to be selected.

An obvious approach to avoid the occurrence of infinite NMSE values would be to collect a big enough training data set, which would hopefully contain a representative sample of all possible clutter sample values. Centres could then be picked from the training data set in a fashion, so as to ensure that they covered the dynamic range of samples in the training data set, and therefore,

[2]To within double precision accuracy on a Sun Sparc 10 workstation.

also of the unseen clutter data. In a practical situation, at sea, this might involve re-training the predictor when changes occurred in environmental conditions. This of course would only be attempted if a NRBFNP could be used to perform better than a LP. If this was not the case, then the simpler LP would be preferred.

A simplistic technique to avoid the occurrence of an infinite NMSE value was investigated for the NRBFNP-RSC example already discussed above, for the prediction of the data in range gate 15. This simplistic technique involved modifying the RSC centres selection method (this new technique will be referred to as the modified-RSC method), so that the training data set was scanned once from start to finish, and any value above a certain threshold (which will be termed the spike threshold) was picked as the starting element of a centre vector: the scanning continued from the end of each centre vector selected in this fashion, until the end of the training data set was encountered. After scanning the training data set once for centres above the spike threshold, centres were picked at random, using the usual RSC method. As with the RSC method, the algorithm for the modified-RSC method was written to ensure that no centres were selected more than once, to avoid ill-conditioning of the data matrix sent to the Householder transform, during training.

Results are now reported for a NRBFNP, with centres selected using the modified-RSC method, for training lengths of 2000 and 6000 samples, using the range gate 15 data set. An embedding dimension of 20, an embedding delay of 1 sample, and 100 Gaussian kernels were used. NRBFNP-RSC results obtained using the same embedding parameters and number of Gaussian kernels are given in Appendix F. The NMSE results for both cases are presented in Table 5.1.

Training length	**Spike threshold**	**Number of spikes**	**Training NMSE [dB]**	**Testing NMSE [dB]**	**Validation NMSE [dB]**
2000	15	1	-6.906503	-2.917879	-3.878289
6000	35	3	-11.831847	-4.025095	-11.119162

Table 5.1: *NRBFNP results for $12ms^{-1}$ range gate 15 data set, using the modified-RSC centres selection technique. An embedding dimension of 20, an embedding delay of 1 sample and 100 Gaussian kernels were used in each case.*

From Table 5.1 it can be seen that for the case with a training length of 2000 samples, the spike threshold in the modified-RSC centres selection method was selected to be 15. This was chosen by visual inspection of the training data set. If it was deemed beneficial, this technique could be improved upon to perhaps use statistical measures of the signal to automate the se-

lection of a suitable spike threshold. However, from the initial prediction analysis described in section 5.2.2, the $12ms^{-1}$ range gate data sets all appeared to be linearly predictable, and therefore such elaboration was not deemed necessary here. Nevertheless, for the case of a training length of 2000 samples, this approach resulted in one centre position being picked, which had a starting element above a value of 15. As can be seen, unlike the results for the corresponding NRBFNP-RSC in Appendix F, no infinite NMSE value was encountered for either the testing or validation data sets. Similar results were obtained for a training length of 6000 samples, which used a spike threshold of 35, which was selected by visual inspection. For both cases of training length, the results reported in Table 5.1 show that infinite NMSE values were avoided, and so NRBFNP failure or breakdown was avoided. However, the simple modified-RSC technique, which was employed to avoid NRBFNP failure, only succeeded because there were large enough spikes in the training data set which made it possible to select at least one centre from, so that the larger spikes in the testing and validation sets did not cause all kernel outputs to be zero. Unfortunately, the spikes in the testing and validation data sets, for the case with a training length of 2000 samples, and in the testing data set, for the case with a training length of 6000 samples, were still that much bigger than the spikes in the training data set to degrade the predictor's performance with respect to the training NMSE. Therefore, spikes present a problem to NRBFNP's. Such spikes also degrade UNRBFNP performance, however, these predictors do not suffer from infinite NMSE values, as no kernel normalisation is performed.

In [14], Haykin also reported that spikes proved a problem for the task of sea clutter prediction. However, the strategy he adopted to deal with spikes must be questioned in view of the claim made by him that clutter is chaotic. Haykin effectively thresholded the training data, so that any training sequences with sea spikes over a certain level were "edited" and not used. There are two consequences to adopting this approach. The first consequence of removing such values from the training phase is that the predictor will not learn to recognise these spikes, and could therefore mistake them for target signals. However, this may be acceptable in a practical situation, where the predictor is part of a detector: false alarms, or mistaking clutter or noise for real targets, is part and parcel of radar target detection systems, the aim is to minimise these false alarms. The second consequence is far more serious for the clutter classification argument Haykin put forward in the same paper. Haykin presented evidence to suggest that sea clutter is a chaotic process, and in fact went so far as to say,

> "...it can be said with confidence that sea clutter is a chaotic process."

However, if a global NLP cannot obtain better prediction performance than a global LP, this would suggest a lack of evidence of underlying *nonlinear* dynamics and thus a lack of evidence of chaos: a discussion along these lines was given in section 4.1. Therefore, if spikes (and thus spiky clutter as a whole) are not found to be nonlinearly predictable (as is the case reported herein and by Haykin [14]), this would suggest that clutter is not a chaotic process. Spikes are an artefact of the clutter signal and may not be regarded as noise or interference, and therefore they may not be simply removed or thresholded from a signal classification perspective.

5.2.4 Further prediction analysis

From the experience gained with the prediction analysis of the Logistic map data and also the Lorenz data, described in sections 4.6 and 4.7, respectively, the best prediction performance in terms of NMSE does not always seem to coincide with a choice of embedding according to Takens' embedding theorem. Whether or not this is true for a wider range of chaotic signals, and indeed, whether or not clutter is actually chaotic at all, it was considered prudent to further investigate the predictability of clutter, using RBFNP's with a wider range of embedding parameters, number of kernels, and prediction steps than already considered. Furthermore, Leung and Lo [12] also adopted a trial and error approach to find the best embedding, and RBFNP parameters for the sea clutter prediction problem.

As discussed in section 5.2.1, most $12ms^{-1}$ range gate data sets had plots of mutual information that were monotonically decreasing, however, a few had plots with discernible minima. Therefore, for further analysis, two range gates were selected, one for each type of mutual information result. Gate 14 was selected as an example of a gate with a monotonically decreasing mutual information plot. Gate 22 was selected as an example of a gate with a mutual information plot with a noticeable minimum value.

Results for range gate 14

Figure 5.5 shows a plot of the time series in range gate 14, along with the mutual information plot for this time series.

Further prediction results for range gate 14 are given in Figure 5.6. Figure 5.6(a) shows NRBFNP-RSC results for NMSE vs number of kernels, where an embedding dimension of 20, an embedding delay of 1 sample, and a training length of 4000 samples were used. LP

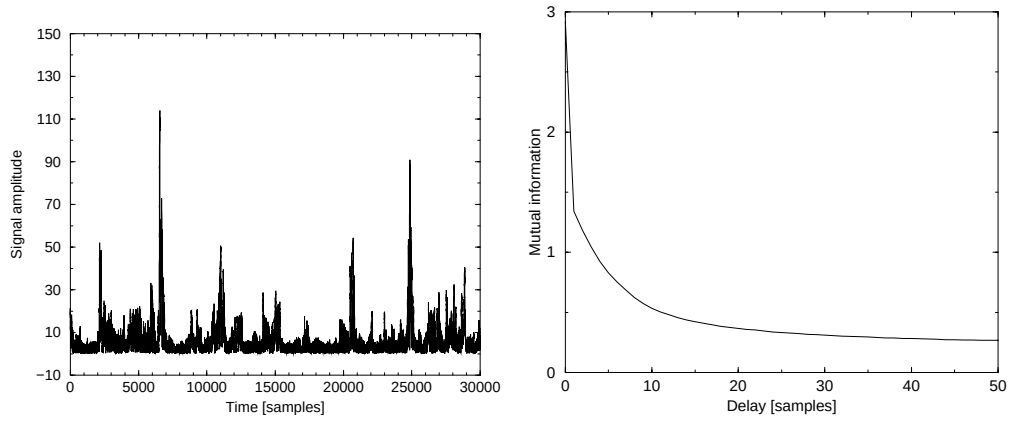


Figure 5.5: *The wavetank $12ms^{-1}$ range gate 14 amplitude data set, (a) time series plot, (b) mutual information plot.*

results are also shown, for a 10 tap LP, which used a training length of 4000 samples. As can be seen in this plot, the simple LP with 10 taps performed better than the NRBFNP-RSC for each number of kernels considered. In fact, increasing the number of kernels above 60, did not add any further NMSE improvement, and for 160 kernels and above, evidence of over-fitting was observed. The reason for the poor performance of the NRBFNP-RSC on the testing data was due to the large spike observed between 4000 and 8000 samples in Figure 5.5(a). The LP did not appear to be adversely affected by this spike.

Figures 5.6(b), 5.6(c), and 5.6(d), show NRBFNP-RSC results for various embedding dimensions, embedding delays and prediction steps[3], respectively. As can be seen in each plot, the LP always performed better than the NRBFNP-RSC, which provides further evidence to suggest that the data is linearly predictable. The reason for the linear predictability of sea clutter is investigated in section 5.4.

Results for range gate 22

Figure 5.7 shows a plot of the time series in range gate 22, along with the mutual information plot for this time series.

Figure 5.7(a) is shown with the same signal amplitude axis range as in Figure 5.5(a). As can be

[3]The prediction step is equivalent to K in equation (4.3).

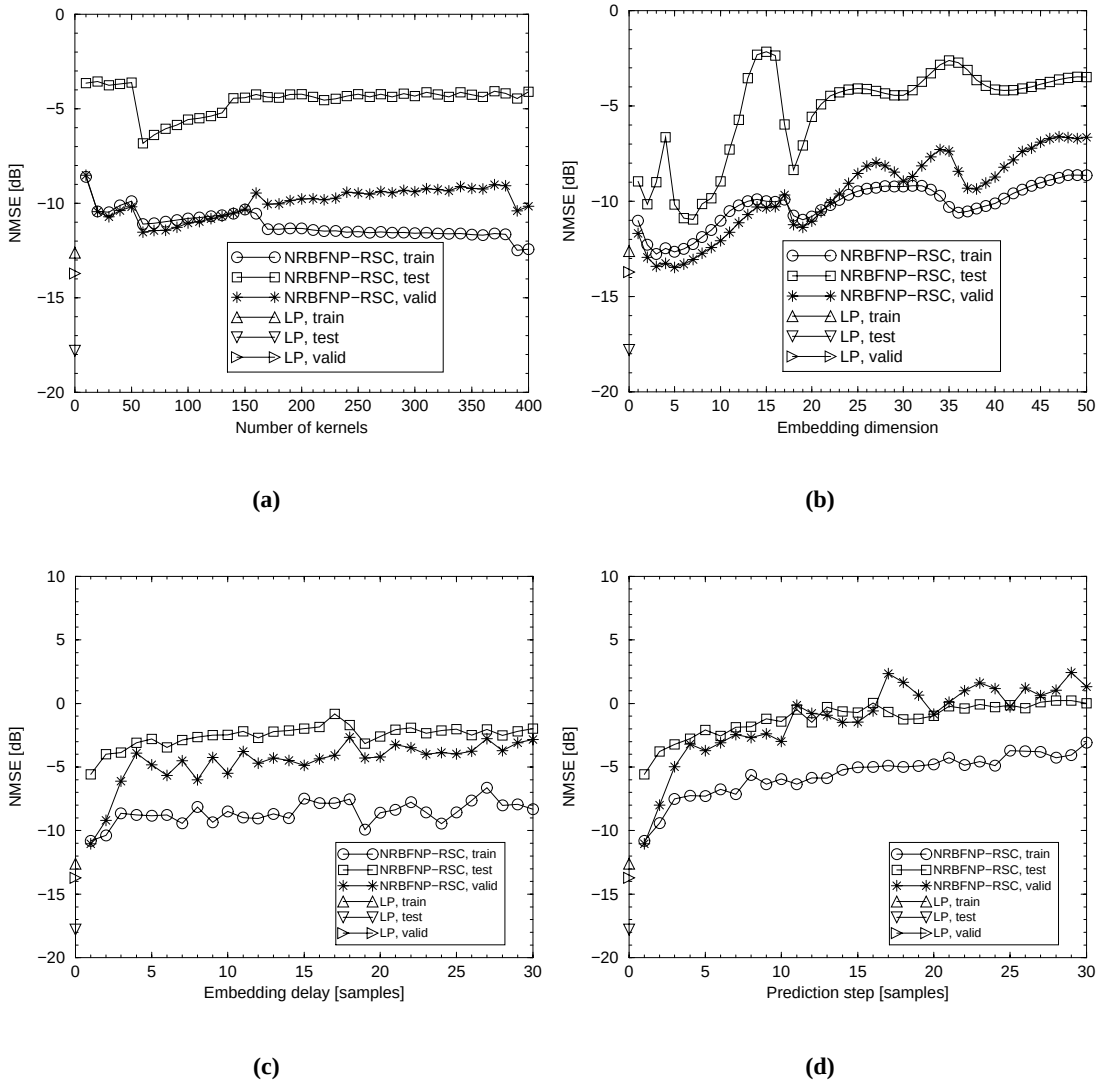


Figure 5.6: *Further prediction analysis of the* $12ms^{-1}$ *range gate 14 data set, using NRBFNP-RSC's with a training length of 4000 samples: (a) NMSE vs number of kernels using an embedding dimension of 20, an embedding delay of 1 sample, and a prediction step of 1 sample, (b) NMSE vs embedding dimension using 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample, (c) NMSE vs embedding delay using an embedding dimension of 20, 100 kernels, and a prediction step of 1 sample, (d) NMSE vs prediction step using an embedding dimension of 20, 100 kernels, and an embedding delay set equal to the prediction step. Linear prediction results are also shown in each plot for a 10 tap LP which had a delay of 1 sample between each tap, a prediction step of 1 sample and a training length of 4000 samples.*

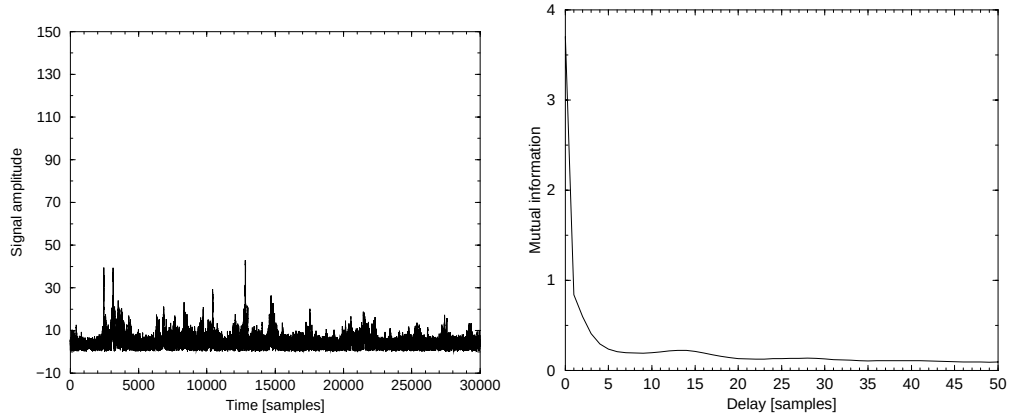


Figure 5.7: *The wavetank $12ms^{-1}$ range gate 22 amplitude data set, (a) time series plot, (b) mutual information plot.*

seen from these two plots, the dynamic range in range gate 14 is far greater than that in range gate 22: the range gate 22 data set is less spiky than the range gate 14 data set. Figure 5.7(b) shows a plot of the mutual information for the data in range gate 22, with a first minimum value at a delay of 9 samples.

Further prediction results for range gate 22 are given in Figure 5.8. Figure 5.8(a) shows NRBFNP-RSC results for NMSE vs number of kernels, where an embedding dimension of 20, an embedding delay of 1 sample, a prediction step of 1 sample, and a training length of 4000 samples were used. LP results are also shown, for a 10 tap LP, which used a training length of 4000 samples. Figure 5.8(b) shows NRBFNP-RSC results for NMSE vs embedding dimension, where 100 kernels, an embedding delay of 1 sample, a prediction step of 1 sample, and a training length of 4000 samples were used. As can be seen from Figures 5.8(a) and 5.8(b), a simple 10 tap LP performed as well as or better than the NRBFNP-RSC's considered. However, unlike the results in Figures 5.6(a) and 5.6(b), where the NRBFNP-RSC's were consistently observed to perform less well than the 10 tap LP, in Figures 5.8(a) and 5.8(b) it can be seen that with a large enough number of kernels and embedding dimension, the NRBFNP-RSC's performed as well as the LP. These results would suggest that the data is linearly predictable as opposed to nonlinearly predictable, and that the NLP's were simply acting as a LP. The fact that the NLP's did as well as the LP with this data set, and not so well with the range gate 14 data set can be attributed to the presence of many large spikes in the range gate 14 data, and fewer spikes

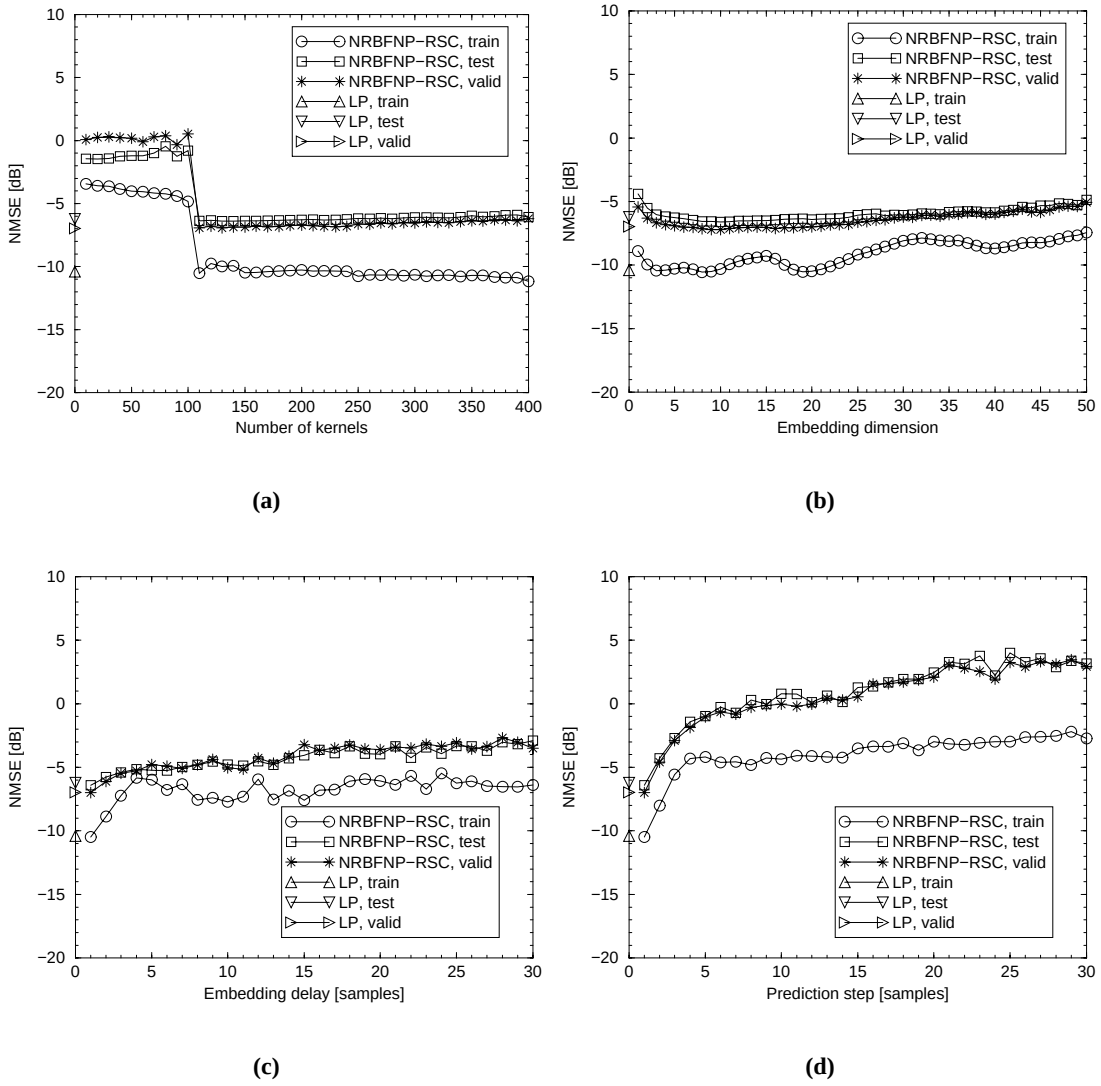


Figure 5.8: *Further prediction analysis of the* $12ms^{-1}$ *range gate 22 data set, using NRBFNP-RSC's with a training length of 4000 samples: (a) NMSE vs number of kernels using an embedding dimension of 20, an embedding delay of 1 sample, and a prediction step of 1 sample, (b) NMSE vs embedding dimension using 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample, (c) NMSE vs embedding delay using an embedding dimension of 20, 100 kernels, and a prediction step of 1 sample, (d) NMSE vs prediction step using an embedding dimension of 20, 100 kernels, and an embedding delay equal to the prediction step. Linear prediction results are also shown in each plot for a 10 tap LP which had a delay of 1 sample between each tap, a prediction step of 1 sample and a training length of 4000 samples.*

in the range gate 22 data set: the presence of spikes in the range gate 14 data set degraded NRBFNP-RSC performance, and this is why it could not perform as well as the LP. NLP's can only perform as well as a LP, if the data they are used on has a linear predictor function. Figures 5.6(c) and 5.6(d), show NRBFNP-RSC results for various different embedding delays and prediction steps, respectively. As can be seen in both plots, the LP always performed at least as well as, or better than the NRBFNP-RSC, which provides further evidence to suggest that the data is linearly predictable. As already mentioned, the reason why the sea clutter has a linear predictor function is investigated in section 5.4.

5.2.5 Prediction of the quadrature channels

Figure 5.9 shows the mutual information plots for the quadrature[4] channel data sets in range gates 14 and 22. As can be seen, the first minimum in the mutual information plots, for all the quadrature channels, in Figure 5.9 occurs at a delay of 4 samples. Compare these mutual information plots with those for the amplitude signals, shown in Figures 5.5 and 5.7 for gates 14 and 22 respectively: the mutual information plot for the amplitude signal in gate 14 was found to be monotonically decreasing, the mutual information for the amplitude signal in gate 22 had a first minimum at a delay of 9 samples. The D_{ML} estimates for the quadrature channels are given in Table 5.2.

Linear and nonlinear prediction was carried out on the quadrature channels of range gates 14 and 22. Nonlinear prediction was carried out using NRBFNP-RSC's. Two different embedding choices were used for the NRBFNP-RSC's: one choice for the embedding parameters was obtained from the D_{ML} values and mutual information plots for the quadrature channels, using the criteria discussed in Chapter 2. The other choice for embedding was the same as that used in the prediction of the corresponding amplitude signals in Figures 5.6 and 5.8. The prediction results are given in Figures 5.10 and 5.11.

The prediction results show that, as with the amplitude data, the quadrature channel data is linearly predictable. In fact, for range gates 14 and 22, the quadrature channel data is slightly less predictable than the amplitude data. The NRBFNP-RSC's were observed to have better generalisation properties on the less spiky gate 22 data, than on the more spiky gate 14 data. However, as the data has been found to be linearly predictable, this observation is moot.

[4] *i.e.* the I (or in-phase) and Q (or quadrature-phase) channel components.

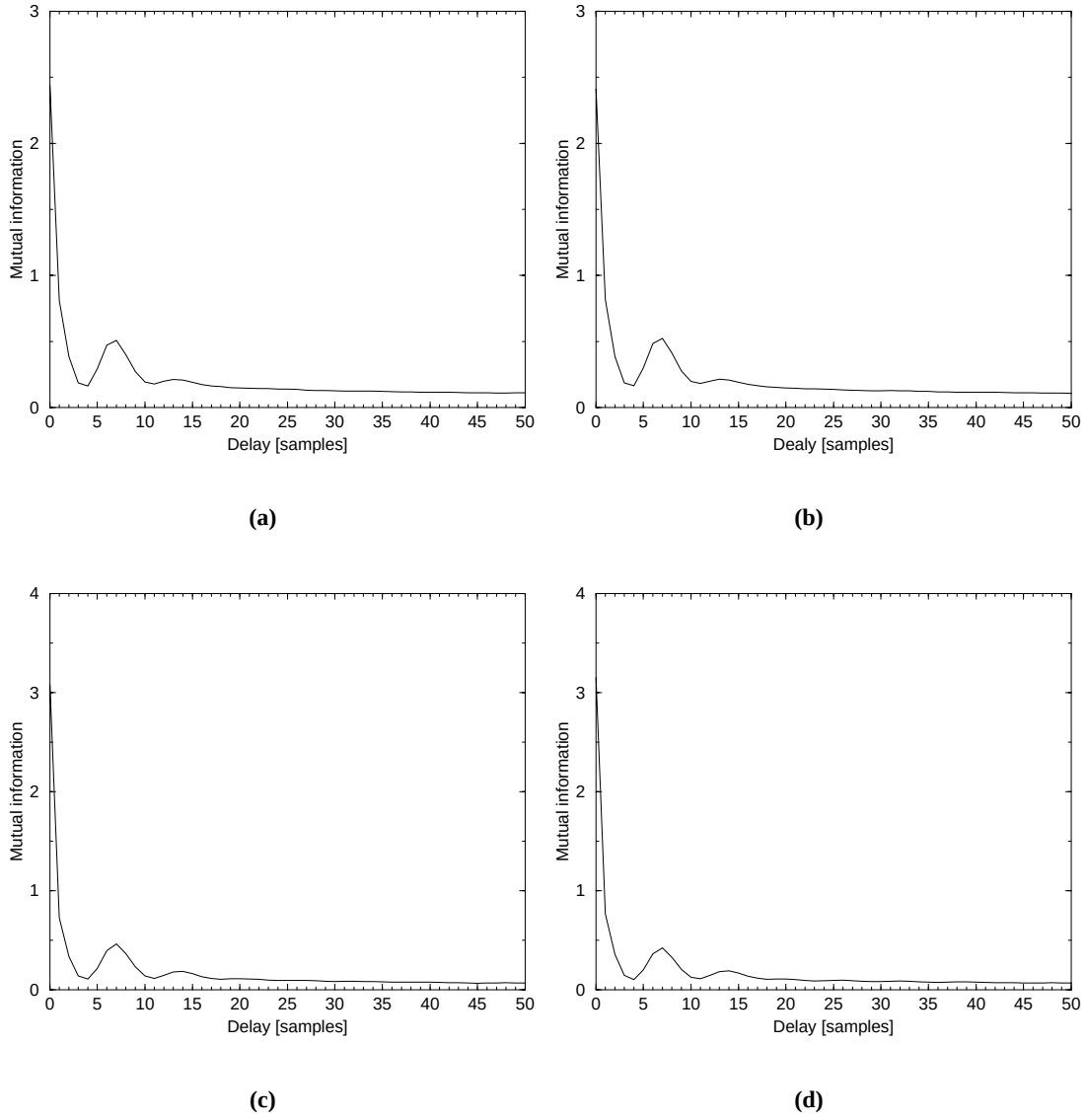
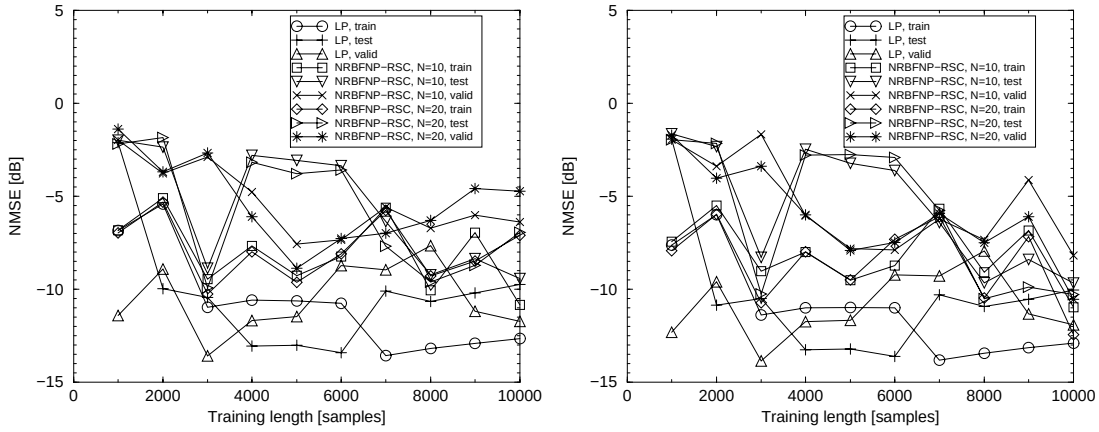


Figure 5.9: *Mutual information plots for the $12ms^{-1}$ range gate 14 (a) I channel, (b) Q channel, and range gate 22 (c) I channel, (d) Q channel.*

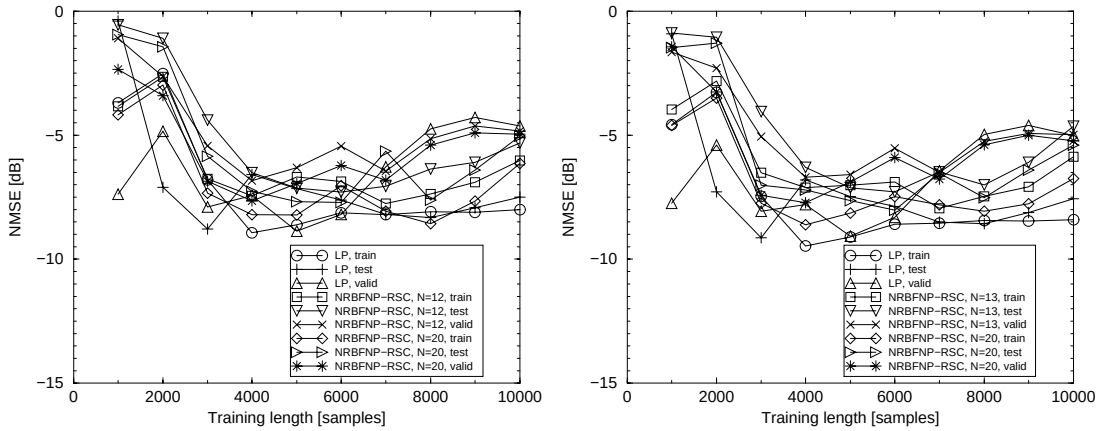
Gate	Channel	Correlation dimension estimate
14	in-phase	4.14815
14	quadrature-phase	4.03896
22	in-phase	5.2027
22	quadrature-phase	5.5244

Table 5.2: *Maximum likelihood correlation dimension estimates D_{ML} for the quadrature channel signals in gate 14 and gate 22.*



(a) **(b)**

Figure 5.10: *Prediction analysis of the $12ms^{-1}$ range gate 14 (a) I and (b) Q channel data sets, using NRBFNP-RSC's with embedding dimensions of 10 and 20. An embedding delay of 4 samples was used with an embedding dimension of 10, and an embedding delay of 1 sample was used with an embedding dimension of 20. For each case of embedding dimension 100 kernels and a prediction step of 1 sample were used. Results are also shown for a 10 tap LP with a delay of 1 sample between each tap, and a prediction step of 1 sample.*



(a) **(b)**

Figure 5.11: *Prediction analysis of the $12ms^{-1}$ range gate 22 (a) I and (b) Q channel data sets, using NRBFNP-RSC's with embedding dimensions of 12 and 20 in (a) and 13 and 20 in (b). An embedding delay of 4 samples was used with embedding dimensions of 12 and 13, and an embedding delay of 1 sample was used with an embedding dimension of 20. For each case of embedding dimension 100 kernels and a prediction step of 1 sample were used. Results are also shown for a 10 tap LP with a delay of 1 sample between each tap and a prediction step of 1 sample.*

5.2.6 Recursive prediction

Recursive prediction was attempted on the amplitude and the in-phase data sets for range gates 14 and 22, as described below.

Recursive prediction was attempted for the gate 14 amplitude, the gate 14 in-phase, the gate 22 amplitude, and the gate 22 in-phase data sets using a LP with 10, 10, 16, and 12 taps respectively. In all cases results were obtained for training lengths of 1000 samples up to 10,000 samples in steps of 1000 samples. A delay of 1 sample between each tap and a prediction step of 1 sample were used for each LP.

Recursive prediction was also attempted for three types of NLP: an UNRBFNP-RSC, a NRBFNP-RSC, and an approximately regularised UNRBFNP-RSC. Each NLP used a prediction step of 1 sample and 500 kernels. An embedding dimension of 10, 10, 16, and 12 were used for the gate 14 amplitude, the gate 14 in-phase, the gate 22 amplitude, and the gate 22 in-phase data sets respectively, for each NLP. An embedding delay of 1 sample, 4 samples, 9 samples, and 4 samples were used for the gate 14 amplitude, the gate 14 in-phase, the gate 22 amplitude, and the gate 22 in-phase data sets respectively, for each NLP. In all cases results were obtained for training lengths of 2000 samples up to 10,000 samples, in steps of 1000 samples. The following range of regularisation parameters were used in the approximately regularised UNRBFNP-RSC, for each training length: $\lambda = 10^0, 10^{-1}, 10^{-2}, \ldots, 10^{-14}$.

Recursive prediction was not successful for any of the above simulations. These results demonstrate a lack of evidence of any underlying nonlinear dynamics for these wavetank clutter data sets. In fact, combined with the results in sections 5.2.2 to 5.2.5, which have shown evidence to suggest that the $12ms^{-1}$ data sets have linear predictor functions, the data appears to be best described as a linear process, contradicting the conclusion reached in [9, 12] that sea clutter has a nonlinear predictor function, and which also contradicts the claim in [11, 13, 14, 17–21] that sea clutter is a chaotic process with underlying *nonlinear* dynamics.

5.3 Prediction of the rest of the wavetank data sets

This section discusses the predictability of the wavetank $4ms^{-1}$, $5ms^{-1}$, $6ms^{-1}$, $7ms^{-1}$, $8ms^{-1}$, $9ms^{-1}$, $10ms^{-1}$, and $11ms^{-1}$ wind speed data sets.

Figure 5.12 shows a plot of prediction results for the $11ms^{-1}$ range gate data sets, using a 10

tap LP, and a cubic VSFP with an embedding dimension of 10, and an embedding delay of 1 sample.

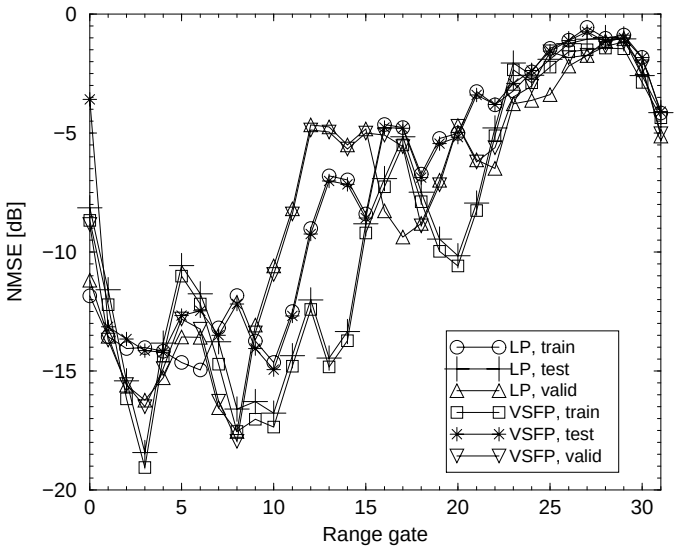


Figure 5.12: *1-step ahead prediction of the $11ms^{-1}$ range gate data sets. Results are shown for a 10 tap LP, and a cubic VSFP with an embedding dimension of 10, and an embedding delay of 1 sample. For each range gate data set, the prediction NMSE result was obtained by using a training length of 10,000 samples, for both the LP, and the VSFP.*

An embedding delay of 1 sample was chosen, after consideration of the nonlinear prediction results already discussed for the $12ms^{-1}$ wavetank data sets, and the nonlinear prediction results in sections 4.6 and 4.7: an embedding delay of 1 sample has been shown to provide the best NMSE results. As can be seen from Figure 5.12, the linear and nonlinear prediction results were very similar. Although the VSFP results were consistently slightly better than the LP on the training phase, the LP and VSFP errors were practically identical on the validation phase: the VSFP could be said to be slightly over-fitting to the training data, but the fact that the LP and VSFP results were similar on the validation data, suggests that the data has a linear predictor function. Furthermore, this analysis was repeated (using a 10 tap LP, and a cubic VSFP with an embedding dimension of 10 and an embedding delay of 1 sample), for the prediction of the $4ms^{-1}, 5ms^{-1}, 6ms^{-1}, 7ms^{-1}, 8ms^{-1}, 9ms^{-1}$, and $10ms^{-1}$ data sets. Again, for these data sets, the LP and VSFP results were very similar: the LP validation error results were always as good as, or better than, the corresponding VSFP results. This evidence suggests that these wind speed data sets also have a linear predictor function.

5.4 Summary of prediction results for the wavetank data

Evidence has been found to suggest that the amplitude and quadrature channel wavetank data sets have a linear predictor function. No evidence of underlying nonlinear dynamics was found using recursive prediction. These results indicate that the wavetank clutter data is not a chaotic process.

Having established the fact that the data is linearly predictable, the variation of predictability with wind speed and across range gates can be illustrated with LP results. Figure 5.13 shows prediction results for the amplitude signal, using a 10 tap LP, for all range gates in each wind speed data set.

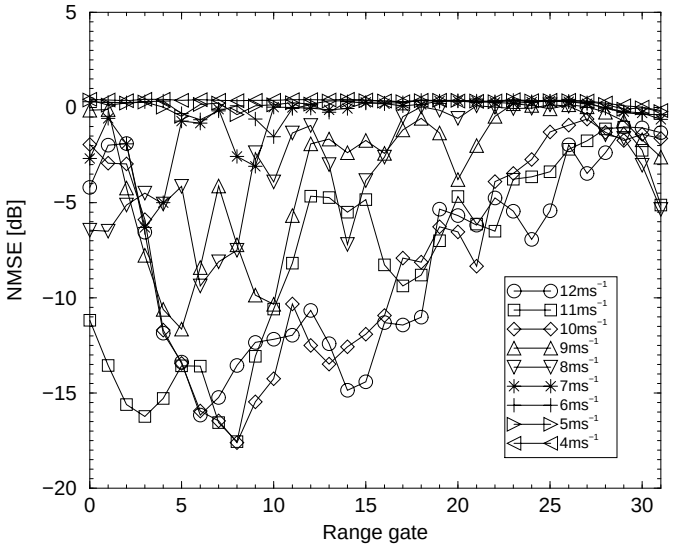


Figure 5.13: *Linear prediction results for wavetank amplitude data sets. Results obtained using a 10 tap LP, with a training length of 10,000 samples, for all range gates in every wind speed data set.*

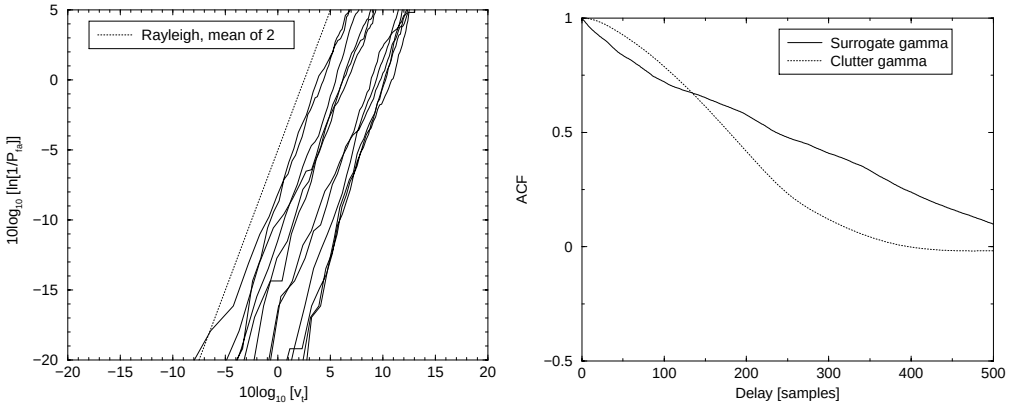
As can be seen from Figure 5.13, in general, the higher wind speed data sets were more predictable than the lower wind speed data sets. There does not appear to be a simple relationship such as the higher the wind speed, the more predictable the data. For instance, some range gates in the $12ms^{-1}$ data set were less predictable than corresponding[5] range gates in lower wind speed data sets. It should be noted that the result, in general, that as windspeed is increased (and

[5]The wavetank data sets were recorded using range gates that were approximately fixed in space: range cells were reported to not move by more than a range cell between different wind speed data recordings. Therefore an exact comparison between range cells for different wind speeds cannot be carried out, however, a reasonable comparison can be.

thus sea state increases) clutter predictability increases, contradicts work done by Haykin and Puthusserypady [21]. Haykin and Puthusserypady noticed that the positive Lyapunov exponent estimates for their sea clutter data sets increased with sea state, and because the magnitude of the largest Lyapunov exponent is inversely proportional to the prediction horizon of a chaotic time series, they stated that this implies that clutter will become less predictable with increasing sea state.

To further investigate the apparent linear predictability of the wavetank sea clutter data sets, and to consider whether or not the compound K-distribution is an appropriate model for these data sets, prediction analysis was carried out using a compound K-distributed surrogate data set of the $12ms^{-1}$ gate 14 amplitude data set.

The process for obtaining compound K-distributed surrogate data sets for any given (sea clutter) amplitude data set, was detailed in Chapter 3. It was found that using a window of size 250 samples (*i.e.* 250ms), the gate 14 amplitude data set was locally Rayleigh distributed. This is illustrated using a Weibull paper plot in Figure 5.14(a).



(a) **(b)**

Figure 5.14: *Obtaining surrogate data for the $12ms^{-1}$ gate 14 data, (a) Weibull paper plot, and (b) gamma component ACF plots for the actual clutter and surrogate data sets.*

To obtain an estimate of the gamma component of the gate 14 amplitude data set, a sliding window of length 250 samples was used to average out the Rayleigh speckle, and produce gamma sample estimates. The shape parameter estimate for these gamma samples was obtained using the method of moments, and was estimated to be 0.1. The scale parameter was estimated to be 0.035. The ACF of these gamma samples was also estimated, and this is shown in Figure

5.14(b) together with the ACF plot of the gamma component of the surrogate clutter data set. As can be seen in Figure 5.14(b), the ACF of the gamma component of the actual clutter data is not accurately represented by a decaying exponential function, which is the method adopted in this thesis for modelling the correlation properties of the gamma component of a compound K-distributed surrogate data set (see Chapter 3). Therefore, the correlation properties of the gamma component of the actual data could only be approximated by the surrogate data set. An IIR filter correlation coefficient of 0.99726 was chosen as a compromise between trying to closely model the first 80 delays of the actual ACF, and avoiding too large a correlation length for the surrogate data. The first 80 delays were recognised as being important[6] as this was found to be the length over which the speckle component had discernible correlations. The correlation properties of the speckle component were estimated, as discussed in Chapter 3, for the first 80 delays. A 4096-long complex FFT (with zero-padding) was used to obtain the weights for the speckle correlation filter $H(w)$, discussed in Chapter 3, using the first 80 delays of the ACF of the speckle component of the actual data. A good fit between the speckle complex ACF's for the actual and surrogate data sets was obtained over the first 20 samples, as is shown in Figure 5.15(a).

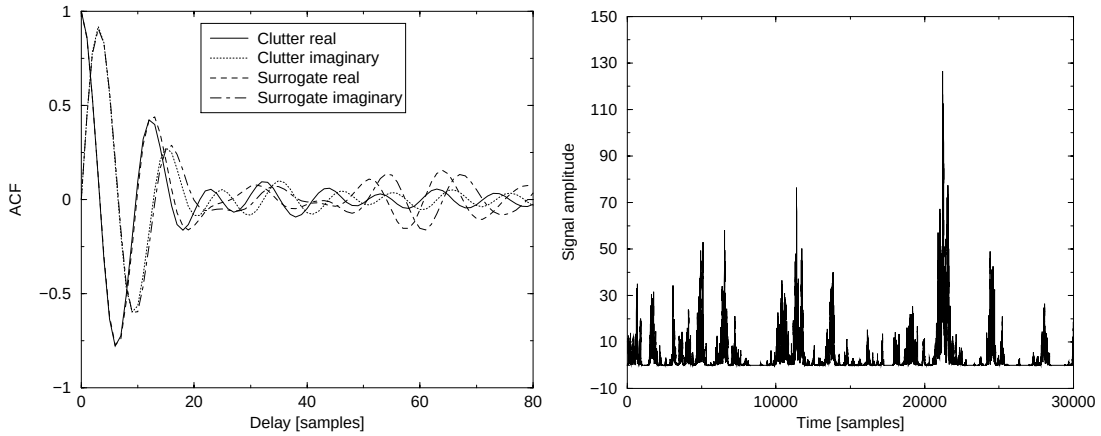


Figure 5.15: *Obtaining surrogate data for the $12ms^{-1}$ gate 14 data, (a) complex ACF of the speckle component of the actual and surrogate clutter data sets, and (b) the amplitude surrogate data set.*

[6]Due to the multiplicative nature of the compound K-distribution simulation technique, and assuming that the speckle and chi (i.e square root of the gamma) components are independent, then the complex ACF of the compound K-distribution is the product of the complex ACF of the speckle component and the chi ACF.

Surrogate amplitude, in-phase, and quadrature-phase data sets were produced for the corresponding $12ms^{-1}$ gate 14 clutter data sets. Figure 5.15(b) shows a plot of the amplitude surrogate data set for gate 14, which can be compared with a plot of the actual data set, given in Figure 5.5(a).

Prediction analysis of the surrogate amplitude data set was carried out using a 10 tap LP, and a NRBFNP-RSC with an embedding dimension of 20, 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample. Results of this prediction analysis are given in Figure 5.16.

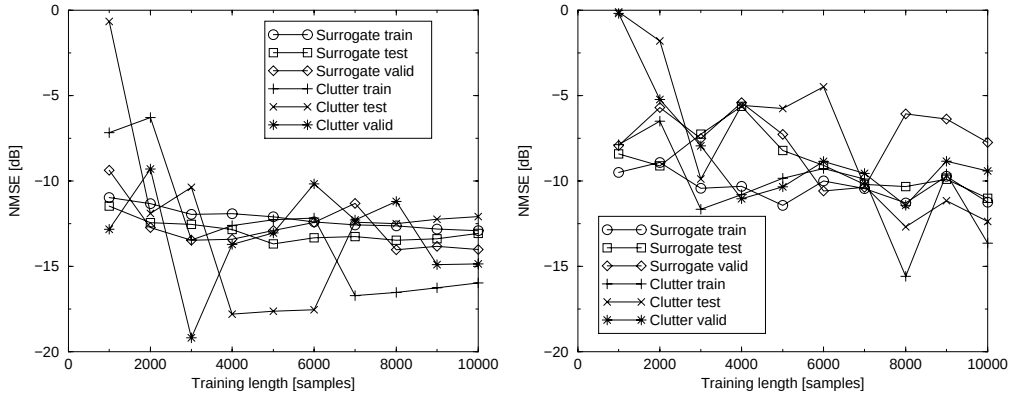
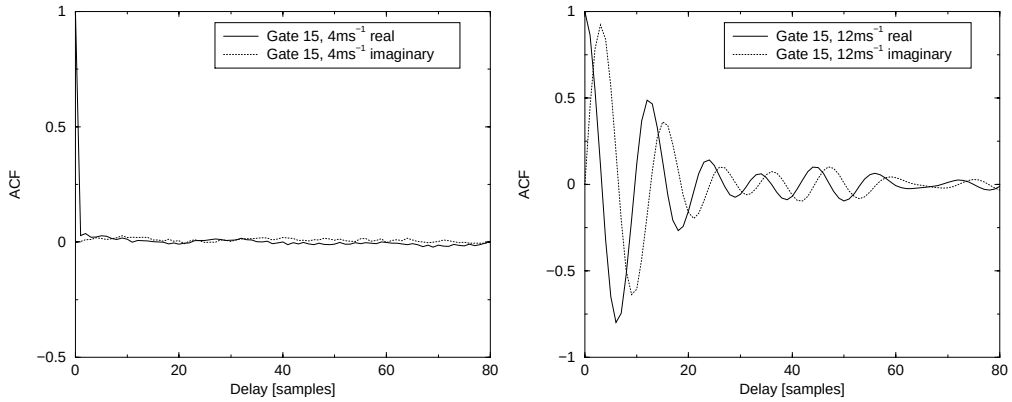


Figure 5.16: *1-step ahead prediction analysis of the $12ms^{-1}$ gate 14 surrogate amplitude data set, (a) linear prediction, (b) nonlinear prediction. A delay of 1 sample was used between each LP tap.*

As can be seen from Figure 5.16, there is a strong similarity between the prediction results for the actual and surrogate clutter data sets. This evidence suggests that the compound K-distribution is an appropriate model for the $12ms^{-1}$ gate 14 data set. Moreover, it is suspected that the (linear) correlations associated with the speckle component of the gate 14 data are the source of its (linear) predictability. If this is the case then the variation in predictability that was illustrated in Figure 5.13, across range for the different wind speed wavetank data sets, can be attributed to corresponding changes in the correlation properties of the speckle component. Further evidence to support this theory is now presented. Figure 5.17 shows the speckle complex ACF plots for range gate 15 from the $4ms^{-1}$, and the $12ms^{-1}$ wind speed data sets. The $4ms^{-1}$ data is uncorrelated, whereas the $12ms^{-1}$ data has a correlated speckle component. As can be seen in Figure 5.13, the $4ms^{-1}$ gate 15 data set was *not* linearly predictable (NMSE was

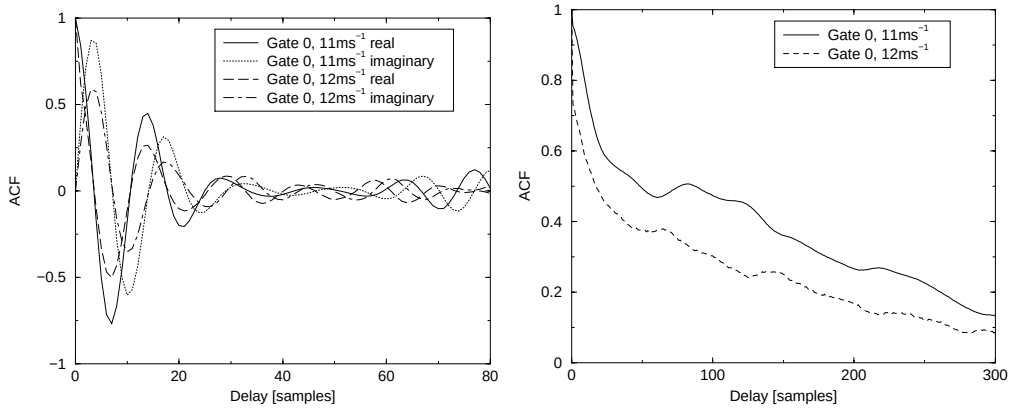


(a) **(b)**

Figure 5.17: *Speckle complex ACF plots for range gate 15 from, (a) the $4ms^{-1}$, and (b) the $12ms^{-1}$ wind speed data sets.*

approximately 0dB), whereas the $12ms^{-1}$ gate 15 data set *was* linearly predictable (NMSE was approximately -15dB).

The influence of speckle correlations is further illustrated by considering the difference between the correlation properties of the $11ms^{-1}$ gate 0 and $12ms^{-1}$ gate 0 data sets, which is shown in Figure 5.18. As can be seen in Figure 5.18(a), the complex speckle component of the $11ms^{-1}$



(a) **(b)**

Figure 5.18: *The difference between the correlation properties of (a) the complex speckle component, and (b) the amplitude component of the $11ms^{-1}$ gate 0 and the $12ms^{-1}$ gate 0 wavetank data sets.*

gate 0 data set is more correlated than that of the $12ms^{-1}$ gate 0 data set, over the first 30 or so

delays. These short-term correlation differences have a related effect on the correlation properties of the corresponding amplitude data sets, and a consequent effect on the predictability of the $11ms^{-1}$ and $12ms^{-1}$ gate 0 amplitude data sets which were reported in Figure 5.13: the NMSE for the $11ms^{-1}$ gate 0 data set was approximately -12dB, whereas the NMSE for the $12ms^{-1}$ gate 0 data set was, approximately, only -4dB.

The above evidence suggests that the wavetank data sets can be well modelled by a compound Gaussian, locally Rayleigh stochastic process, even if the overall process is not K-distributed. For example, Noga [39] found that the underlying component of the sea clutter he analysed was better modelled by a lognormal distribution than by the gamma distribution.

The prediction results for the wavetank data sets are now summarised.

- The amplitude and quadrature channel data sets were found to be linearly predictable
- A compound Gaussian stochastic model was found to be a suitable model
- The linear predictability was found to be associated with the correlation properties of the speckle component of the compound Gaussian model
- The variation in predictability across range gates, and/or, between windspeed data sets was due to a corresponding variation in the correlation properties of the speckle component

To establish the factors that cause a variation in the correlation properties of the speckle component, a more thorough experimental set-up would be required. For instance, fixed range gates would need to be used for the different windspeed conditions, and quantities such as sea surface wave height and wavelength (with respect to the range gate positions) should be recorded to see if there is any correlation between these measurements, and the variation of predictability across range.

At this point it should be noted that although the results presented for the wavetank data sets appear to suggest that the data sets are linearly predictable, and are best modelled by a stochastic (locally Rayleigh) compound model, such as the compound K-distribution, rather than as a chaotic process, there are two important differences between the collection of the wavetank data and the collection of the data analysed by the researchers who found evidence to suggest that clutter is nonlinearly predictable and chaotic. The first difference is that polarisation agility was

used in the collection of the wavetank data, but not by the researchers who found evidence that clutter is chaotic. However, this is not considered significant, as this agility was not randomised, but applied in a known sequence. The second difference is that pulse compression [43, 126] was used in the collection of the wavetank data, but not used by the researchers who found evidence of chaotic behaviour in their sea clutter data sets. Pulse compression is a technique used to achieve a higher resolution in range (*i.e.* a shorter transmitted pulse width). It may be possible that the operation of pulse compression could mask or remove any evidence of nonlinear predictability, and of chaotic behaviour. Furthermore, the area illuminated on the sea surface is proportional to the width of the uncompressed pulse, and not to the width of the compressed pulse. To further investigate whether or not pulse compression is responsible for the removal of the nonlinear predictability of clutter, it is suggested that experiments be carried out using two radars: one which employs pulse compression to obtain sea clutter data at a certain range resolution, the other which does not employ pulse compression to obtain clutter at the same resolution.

Apart from the difference in radar parameters used to collect the wavetank data, it should also be pointed out that the wavetank in which the data sets were collected might not accurately represent conditions at sea. For example, the effects of gravity waves on the collection of sea clutter may not be well modelled in the experimental set-up of the wavetank. This could be a further reason why no chaotic behaviour has been detected. However, to counter this argument, the popular compound K-distribution model of sea clutter provides a suitable model for the wavetank data, suggesting that perhaps conditions at sea have been represented accurately.

One final note, to complement the above comparison of data sets, is to make the following observation. The data sets analysed in [9, 12–18], for evidence of nonlinear predictability and of chaotic dynamics (using recursive prediction) all appeared to be much less spiky than the wavetank data analysed in this thesis. This is despite spiky data being available to the researchers whose work is referenced in [9, 12–18]. Spiky data sets, with similar time histories to some of the wavetank data sets were presented in [62]. These data sets were collected using the IPIX radar facility which is available to the researchers whose work is referenced in [9, 12–18]. This discussion is also relevant to the observations made about the detrimental effect spikes were observed to have on a NRBFNP, as discussed in section 5.2.

5.5 Prediction of the Dawber data sets

This section discusses the prediction of the Dawber amplitude and quadrature channel data sets, which are discussed in Appendix A.

5.5.1 Time series plots, mutual information plots, and correlation dimension estimates for the amplitude data sets

Time series and mutual information plots for the Dawber amplitude data sets are given in Figure 5.19.

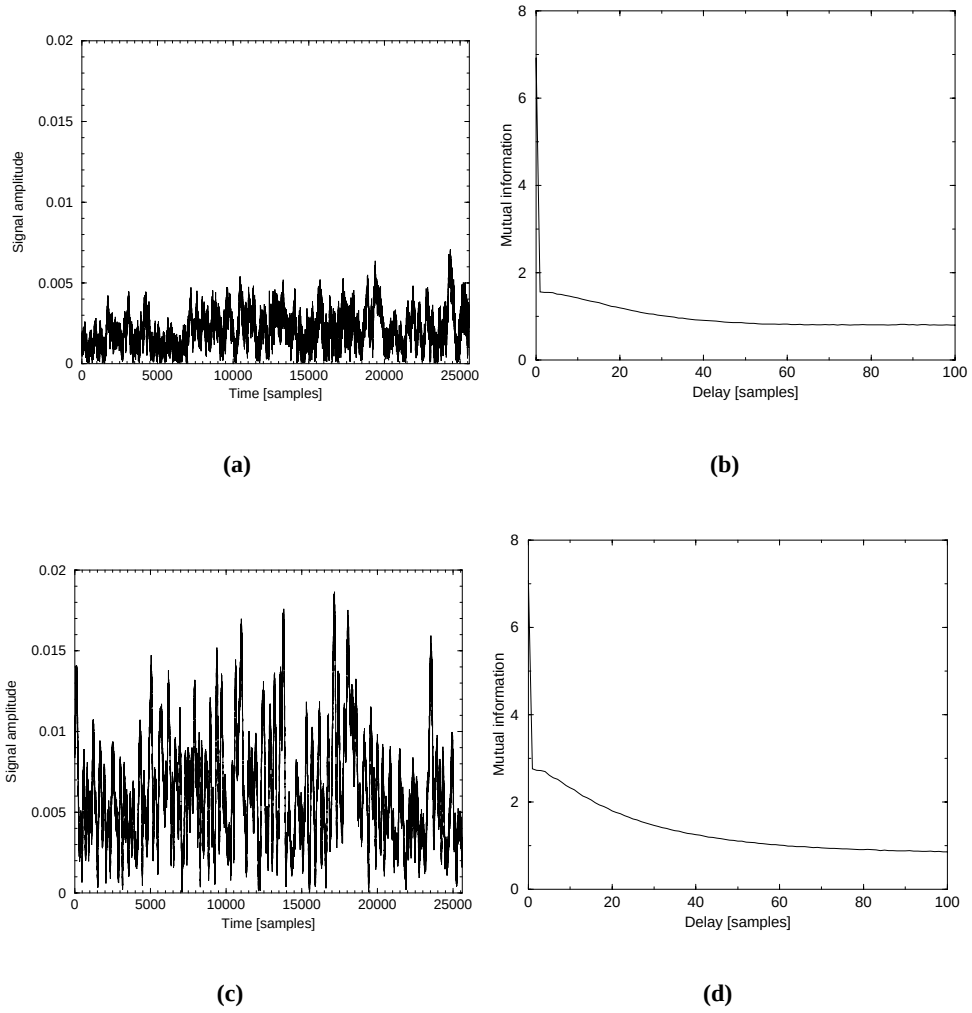


Figure 5.19: *The Dawber HH amplitude data, (a) time series plot, (b) mutual information plot, and the Dawber VV amplitude data, (c) time series plot, (d) mutual information plot.*

The mutual information plots for both Dawber data sets were found to be monotonically decreasing. Table 5.3 shows the correlation dimension estimates for the Dawber amplitude data sets.

Data set	Correlation dimension
HH	7.46876
VV	3.45714

Table 5.3: *Maximum likelihood correlation dimension D_{ML} estimates for the Dawber amplitude sea clutter data sets.*

5.5.2 Prediction of the amplitude data

Initial prediction results, see Figure 5.20, for the Dawber amplitude data sets gave evidence which suggested that these data sets have a linear predictor function. An embedding delay of 1 sample was used for this analysis, as the mutual information plots in Figure 5.19 were found to be monotonically decreasing, and an embedding delay of 1 sample has been shown to be the optimal choice for the prediction of the signals so far considered in this thesis.

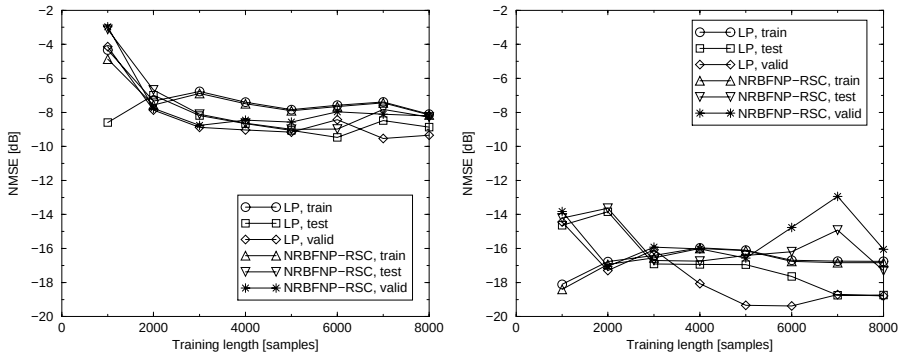


Figure 5.20: *Initial 1-step ahead prediction analysis of the (a) Dawber HH and (b) Dawber VV amplitude data sets, using NRBFNP-RSC's with 100 kernels and an embedding delay of 1 sample. An embedding dimension of 17 was used for the analysis in (a), whilst an embedding dimension of 10 was used for the analysis in (b). Results are also shown for a 10 tap LP.*

NRBFNP-RSC prediction results were recorded for a range of embedding parameters, number of kernels and prediction steps, to determine if a NRBFNP-RSC could outperform a LP. These results are presented in Figure 5.21.

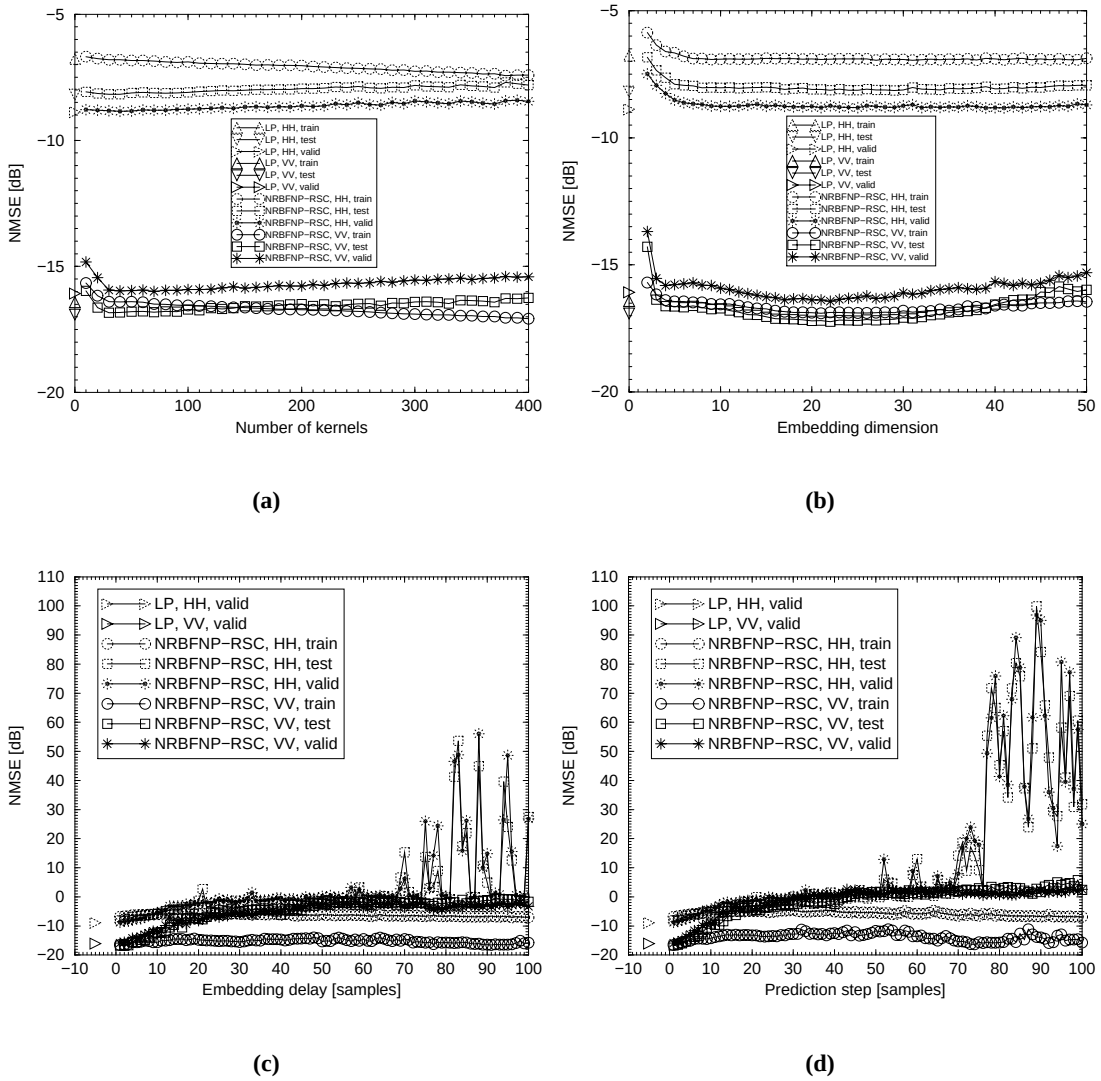


Figure 5.21: *Further prediction analysis of the Dawber amplitude data sets, using NRBFNP-RSC's with a training length of 3000 samples: (a) NMSE vs number of kernels using an embedding dimension of 17 for the HH data and 10 for the VV data, with an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (b) NMSE vs embedding dimension using 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (c) NMSE vs embedding delay using an embedding dimension of 17 for the HH data and 10 for the VV data, with 100 kernels, and a prediction step of 1 sample for both data sets, (d) NMSE vs prediction step using an embedding dimension of 17 for the HH data and 10 for the VV data, with 100 kernels, and an embedding delay equal to the prediction step for both data sets. Linear 1-step ahead prediction results are also shown in each plot for a 10 tap LP (with a 1 sample delay between each tap input) which had a training length of 3000 samples.*

Results are also presented for a simple LP with 10 taps. The results in Figure 5.21 show that for a number of different parameter combinations, the performance of the simple 10 tap LP was always as good as, or better than, a NRBFNP-RSC, for both the HH and VV data sets. This evidence suggests that both data sets have a linear predictor function. Also note from Figure 5.21, that the HH data was less predictable than the VV data. A discussion of the source of the predictability of the Dawber data sets is given in section 5.5.5.

Note that in Figures 5.21(c) and 5.21(d), some very large positive NMSE values were encountered. These values were encountered due to the ill-conditioning of the data matrix. Ill-conditioning was mentioned in section 4.4.5, and as also mentioned in section 4.4.5, singular value decomposition can be used instead of the Householder transform to avoid the problems associated with an ill-conditioned data matrix, such as the spurious large positive NMSE values mentioned above. This is demonstrated in Figure 5.22.

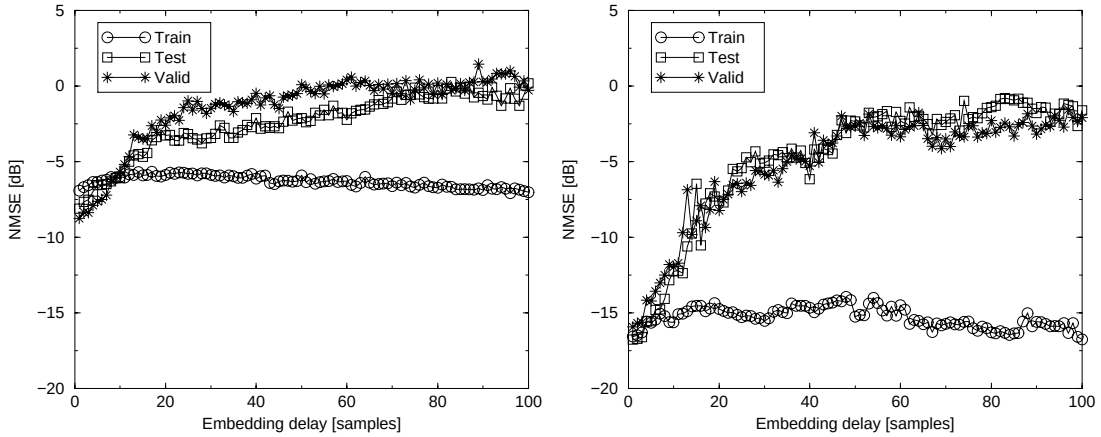


Figure 5.22: *Using singular value decomposition to solve for a NRBFNP-RSC's output layer weights when the data matrix is ill-conditioned. Results are shown in (a) for the HH amplitude data and in (b) for VV amplitude data. NMSE vs embedding delay results obtained using a NRBFNP-RSC with an embedding dimension of 17 for the HH data and 10 for the VV data, with 100 kernels, and a prediction step of 1 sample were used for both data sets.*

The results in Figure 5.22 confirm that the spurious results obtained in Figure 5.21(c) were due to ill-conditioning. The spurious results in Figure 5.21(d) were also attributed to ill-conditioning.

5.5.3 Time series plots, mutual information plots, and correlation dimension estimates for the quadrature channel data sets

Time series and mutual information plots for the in-phase HH and VV data sets are given in Figure 5.23.

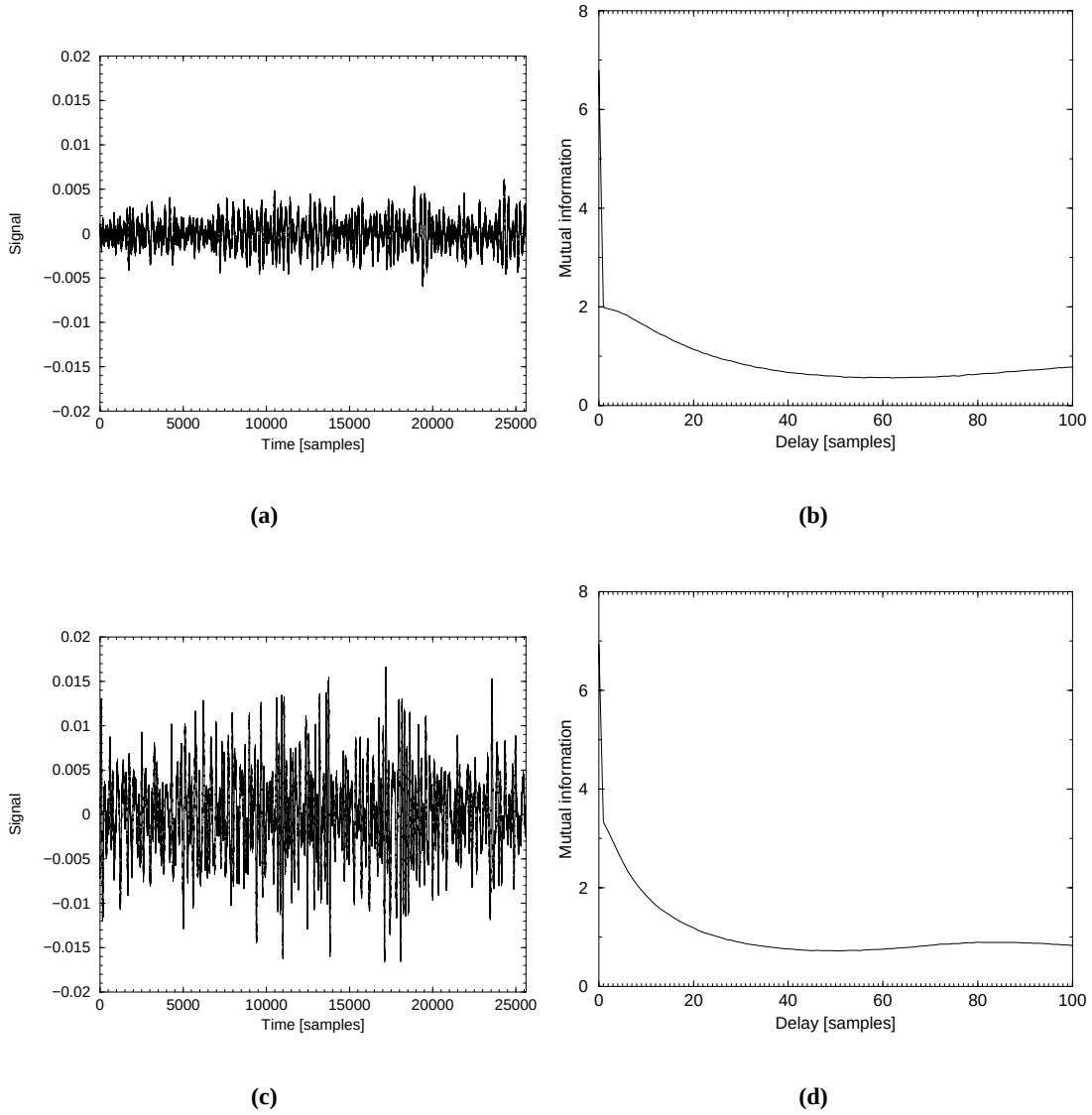


Figure 5.23: *The Dawber HH in-phase data, (a) time series plot, (b) mutual information plot, and the Dawber VV in-phase data, (c) time series plot, (d) mutual information plot.*

A minimum was discernible in both mutual information plots, at a delay of 57 samples for the HH in-phase channel mutual information plot, and for a delay of 50 samples for the VV

in-phase channel mutual information plot. Correlation dimension estimates for the Dawber quadrature channel data sets are given in Table 5.4.

Data set	Correlation dimension
HH in-phase	5.70967
HH quadrature-phase	5.70967
VV in-phase	3.47368
VV quadrature-phase	3.45454

Table 5.4: *Maximum likelihood correlation dimension D_{ML} estimates for the Dawber quadrature channel sea clutter data sets.*

5.5.4 Prediction of the quadrature channels

Initial prediction results, see Figure 5.24, for the Dawber in-phase channel data sets gave evidence which suggested that these data sets have a linear predictor function. An embedding delay of 1 sample was used for this analysis, as this has been shown to be the optimal choice for the prediction of the signals so far considered in this thesis.

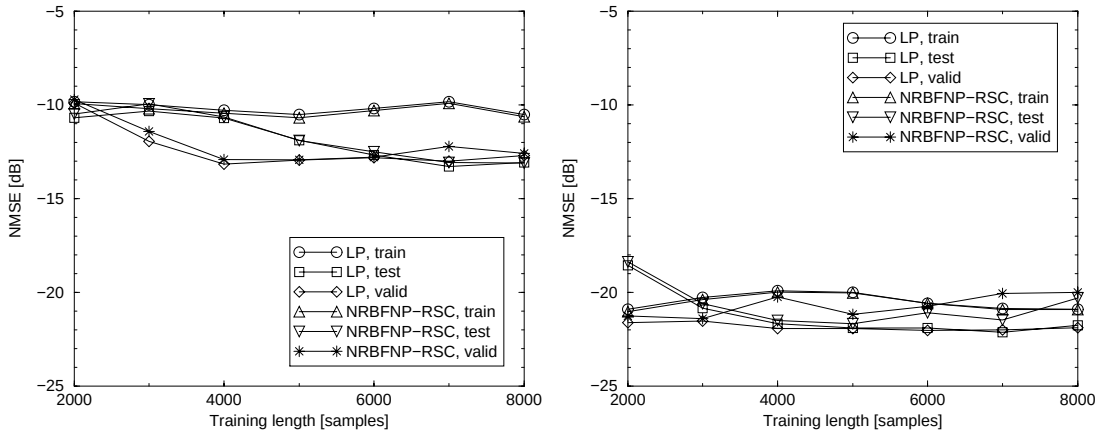


Figure 5.24: *Initial 1-step ahead prediction analysis of the (a) Dawber HH and (b) Dawber VV in-phase data sets, using NRBFNP-RSC's with 100 kernels and an embedding delay of 1 sample. An embedding dimension of 14 was used for the analysis in (a), whilst an embedding dimension of 10 was used for the analysis in (b). Results are also shown for a 10 tap LP.*

In Figure 5.25 NRBFNP-RSC results are shown for a range of embedding parameters, number of kernels, and prediction steps.

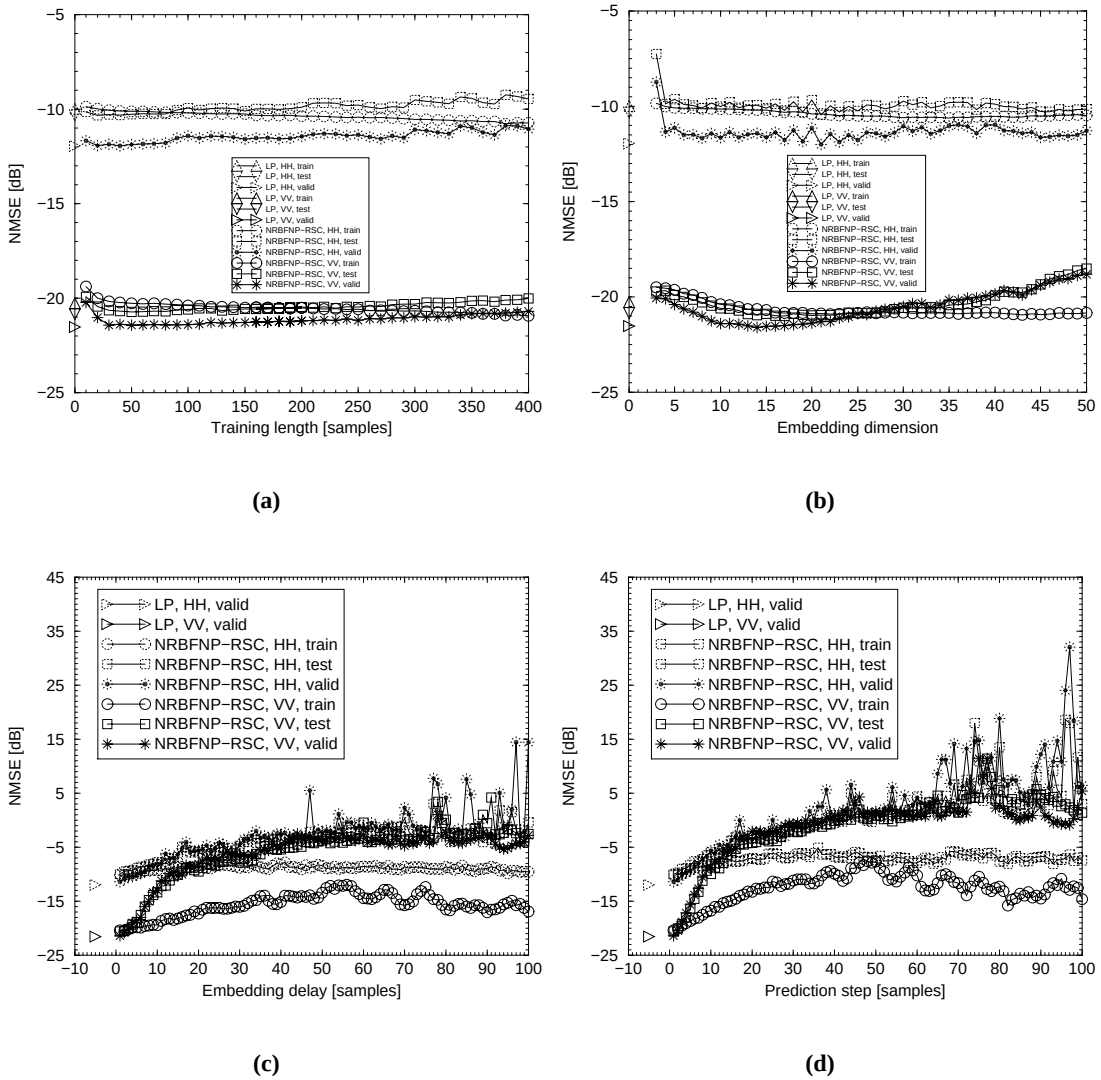


Figure 5.25: *Further prediction analysis of the Dawber in-phase data sets, using NRBFNP-RSC's with a training length of 3000 samples: (a) NMSE vs number of kernels using an embedding dimension of 14 for the HH data and 10 for the VV data, with an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (b) NMSE vs embedding dimension using 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (c) NMSE vs embedding delay using an embedding dimension of 14 for the HH data and 10 for the VV data, with 100 kernels, and a prediction step of 1 sample for both data sets, (d) NMSE vs prediction step using an embedding dimension of 14 for the HH data and 10 for the VV data, with 100 kernels, and an embedding delay equal to the prediction step for both data sets. Linear 1-step ahead prediction results are also shown in each plot for a 10 tap LP (with a 1 sample delay between each tap input) which had a training length of 3000 samples.*

Results are also presented for a simple LP with 10 taps. The results in Figure 5.25 show that for a range of different parameter combinations, the performance of the simple 10 tap LP was always as good as, or better than a NRBFNP-RSC, for both the HH and VV in-phase channel data sets. This evidence suggests that both data sets have a linear predictor function. The spurious large values of NMSE in Figures 5.25(c) and 5.25(d) are a result of ill-conditioning, as discussed in section 5.5.2.

5.5.5 Surrogate prediction

To further investigate the apparent linear predictability of the Dawber sea clutter data sets, prediction analysis was carried out using compound K-distributed surrogate data sets of the Dawber VV data sets.

The process for obtaining a compound K-distributed surrogate data set for any given (sea clutter) data set, was detailed in Chapter 3. It was found that using a window of size 5000 samples (*i.e.* 250ms), the Dawber VV amplitude data set was locally Rayleigh distributed. This is illustrated using a Weibull paper plot in Figure 5.26(a).

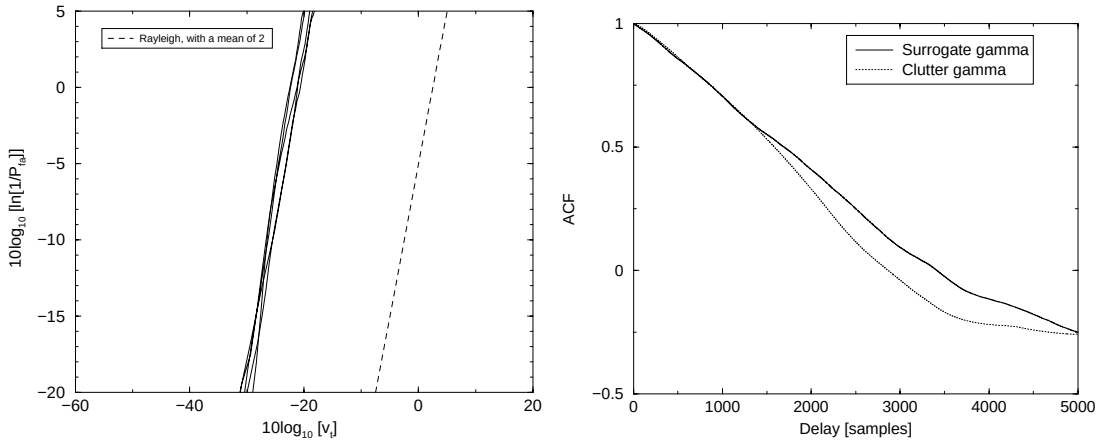


Figure 5.26: *Obtaining surrogate data for the Dawber VV data, (a) Weibull paper plot, and (b) gamma component ACF plots for the actual and surrogate clutter data sets.*

To obtain an estimate of the gamma component of the Dawber VV amplitude data set, a sliding window of length 5000 samples was used to average out the Rayleigh speckle, and produce gamma sample estimates. The shape parameter estimate for these gamma samples was obtained

using the method of moments, and was estimated to be 23. The scale parameter was estimated to be 661. The ACF of these gamma samples was also estimated, and this is shown in Figure 5.26(b) together with the ACF plot of the gamma component of the surrogate clutter data set. As can be seen in Figure 5.26(b), the ACF of the gamma component of the actual clutter data is not accurately represented by a decaying exponential function, which is the method adopted in this thesis for modelling the correlation properties of the gamma component of a compound K-distributed surrogate data set (see Chapter 3). Therefore, the correlation properties of the gamma component of the actual data could only be approximated by the surrogate data set. An IIR correlation coefficient of 0.99979999 was chosen as a compromise between trying to closely model the first 440 delays of the actual ACF, and avoiding too large a correlation length for the surrogate data. The first 440 delays were recognised as being important as this was found to be the length over which the speckle component had discernible correlations. The correlation properties of the speckle component were estimated, as discussed in Chapter 3, for the first 440 delays. A 4096-long complex FFT (with zero-padding) was used to obtain the weights for the speckle correlation filter $H(w)$, discussed in Chapter 3, using the first 440 delays of the ACF of the speckle component of the actual data. A reasonable fit between the speckle complex ACF's for the actual and surrogate data sets was obtained over the first 100 samples, as is shown in Figure 5.27(a).

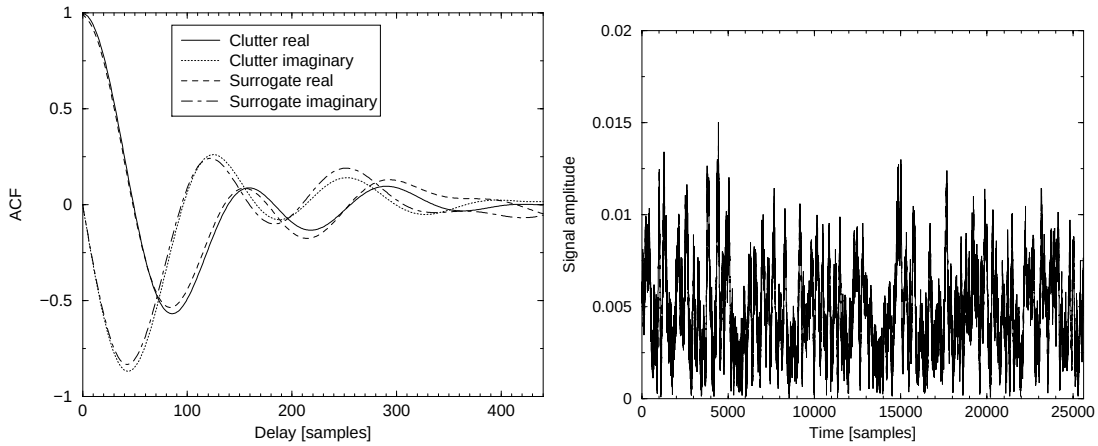


Figure 5.27: *Obtaining surrogate data for the Dawber VV data, (a) speckle complex ACF of the actual and surrogate clutter data sets, and (b) the amplitude surrogate data set.*

Surrogate amplitude, in-phase, and quadrature-phase data sets were produced for the corres-

ponding Dawber VV clutter data sets. Figure 5.27(b) shows a plot of the amplitude surrogate data set, which can be compared with a plot of the actual data set, given in Figure 5.19(c).

Prediction analysis of the surrogate amplitude data set was carried out using a 10 tap LP, and a NRBFNP-RSC with an embedding dimension of 10, 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample. Results of this prediction analysis are given in Figure 5.28.

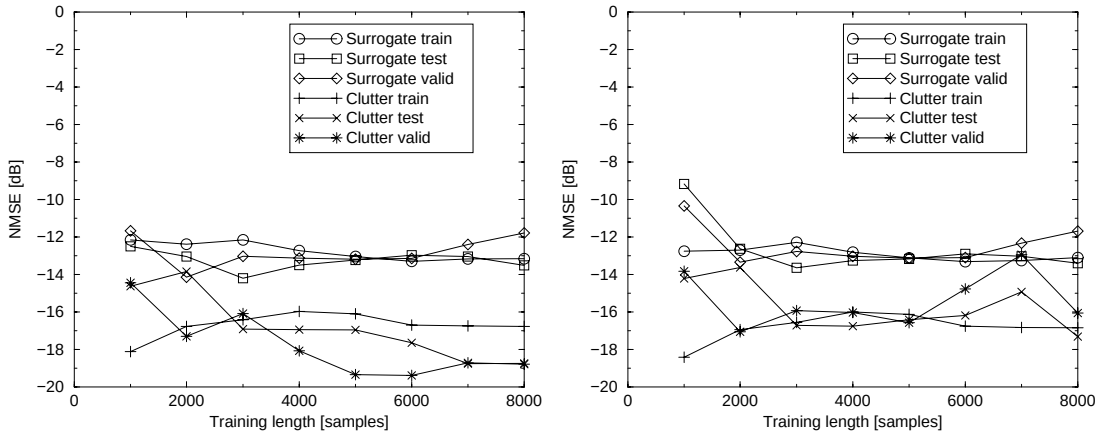


Figure 5.28: *1-step ahead prediction analysis of the Dawber VV surrogate amplitude data set, (a) linear prediction, (b) nonlinear prediction.*

As can be seen from Figure 5.28, the surrogate data set was found to be less predictable than the actual clutter data set. At first it might be thought that this implies that the compound K-distribution is not an appropriate model for the Dawber VV data. However, from the evidence in Figure 5.26(a) it would appear that the Dawber VV data is at least a locally Rayleigh (*i.e.* a compound Gaussian) process. It may be that the mean level of the locally Rayleigh component of the Dawber VV data does not have a chi-distributed mean level, and therefore the compound K-distribution would not be an appropriate model for the data: for example Noga [39] found that a locally Rayleigh process with a lognormal distributed mean level better fitted the sea clutter available to him, than did a locally Rayleigh process with a chi-distributed mean level. However, the high PRF (20 times that used for the collection of the wavetank data), and the short data record (less than the number of samples collected in each range cell of the wavetank data sets), are thought to be responsible for the inability to obtain a reasonable estimate for the average complex ACF of the speckle component of the Dawber data. This poor estimation of

the speckle ACF is thought to be the reason why there is a lack of correspondence between the prediction results for the actual and surrogate data sets. Therefore, evidence has not been presented to suggest that the compound K-distribution is an appropriate model for the Dawber VV data, although evidence has been presented which suggests that the data can be modelled as a compound Gaussian process. In which case, the linear predictability of the Dawber data sets discussed in sections 5.5.2 and 5.5.4 can be associated with Gaussian correlations.

5.5.6 Recursive prediction

Recursive prediction was initially carried out on the in-phase Dawber VV data set, as this was found to be the most predictable data set (of all the Dawber amplitude and quadrature channel data sets), and it had a discernible mutual information minimum value.

Recursive prediction was attempted for the Dawber VV in-phase data set using the following predictors. A LP with 10 taps, a 1 sample delay between each tap, and a prediction step of 1 sample, for training lengths of 1000 up to 10,000 samples, in steps of 1000 samples. Additionally, LP simulations were run with exactly the same parameters as those above, except a delay of 50 samples was used between each tap. Three types of NLP were used: an UNRBFNP-RSC, a NRBFNP-RSC and an approximately regularised UNRBFNP-RSC. Each NLP used an embedding dimension of 10, an embedding delay of 50 samples and a prediction step of 1 sample. These embedding parameters were chosen using the criteria given in Chapter 2. Each NLP used 500 kernels. Recursive prediction results were obtained for the NLP's using training lengths of 2000 up to 10,000 samples, in steps of 1000 samples. The following range of regularisation parameters were used in the approximately regularised UNRBFNP-RSC, for each training length: $\lambda = 10^{0}, 10^{-1}, 10^{-2}, ..., 10^{-14}$.

Recursive prediction was not successful for any of the LP, UNRBFNP-RSC or approximately regularised UNRBFNP-RSC simulations. Recursive prediction was successful for 2 NRBFNP-RSC simulations: for training lengths of 6000 and 8000 samples. However, although recursive prediction worked, the recursively predicted signal clearly did not represent the original Dawber VV in-phase data set. This is illustrated in Figure 5.29.

As some success was encountered for recursive prediction with the embedding criteria of Chapter 2, it was decided that a trial and error approach (towards the selection of the embedding parameters of a NRBFNP-RSC) would be used to try and find out if a NRBFNP-RSC

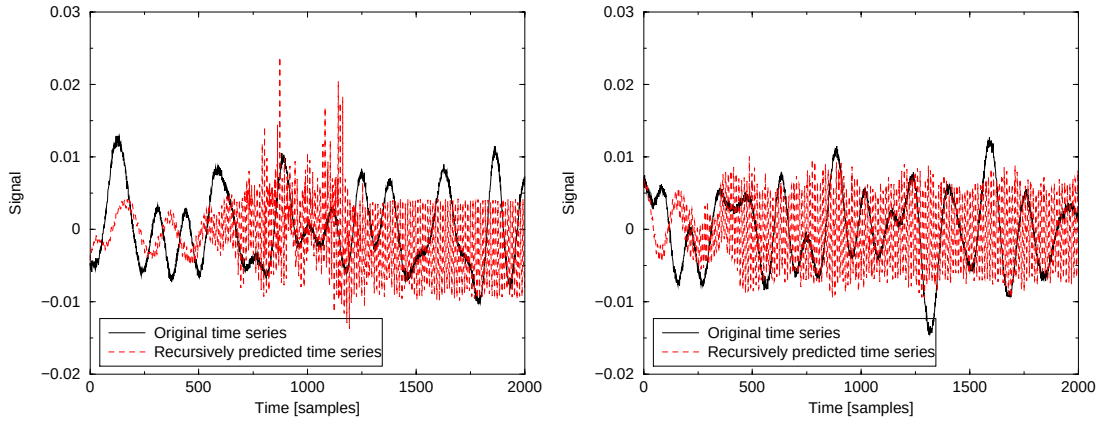
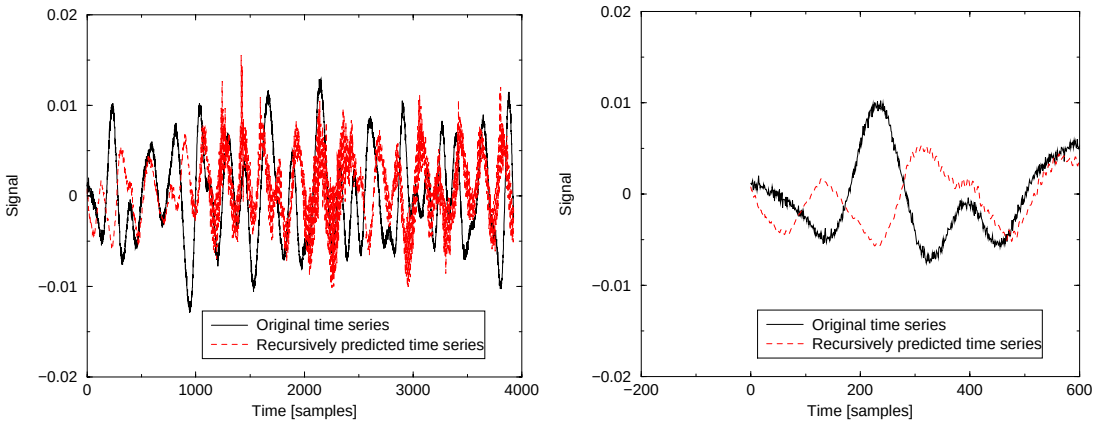


Figure 5.29: *Recursive prediction of the Dawber VV in-phase data set. Results are shown for a NRBFNP-RSC with an embedding dimension of 10, an embedding delay of 50 samples, and prediction step of 1 sample, for 2 different training lengths: (a) 6000 samples, and (b) 8000 samples.*

could achieve better recursive prediction results than those illustrated in Figure 5.29. NRBFNP-RSC simulations were run with an embedding dimension of 10, 500 kernels, a prediction step of 1 sample, and embedding delays of 1 sample, 2 samples,...,10 samples. For each different embedding delay, simulations were run for training lengths of 1000 samples up to 10,000 samples in steps of 1000 samples. Promising recursive prediction results were obtained for an embedding delay of 8 samples, and a training length of 4000 samples. This result is shown in Figure 5.30.

The recursively predicted time series shown in Figure 5.30(a) looks similar to the actual time series, and it certainly is an improvement on the results obtained in Figure 5.29. Upon closer inspection, see Figure 5.30(b), it can be seen that there is no prediction horizon associated with the recursively predicted time series, *i.e.* the original and recursively predicted time series diverge from the first recursively produced sample.

Using the NRBFNP-RSC with an embedding delay of 10, 500 kernels, an embedding delay of 8 samples, a prediction step of 1 sample, and a training length of 4000 samples, 25600 (the length of the Dawber data sets) samples were recursively predicted. From this recursively produced time series, the mutual information was plotted, see Figure 5.31, and the maximum likelihood correlation dimension was estimated as 4.50538.



(a) **(b)**

Figure 5.30: *Recursive prediction of the Dawber VV in-phase data set. Results are shown for a NRBFNP-RSC, with an embedding dimension of 10, 500 kernels an embedding delay of 8 samples, and prediction step of 1 sample, for a training length of 4000 samples: (a) first 4000 recursively predicted samples, (b) first 600 recursively predicted samples.*

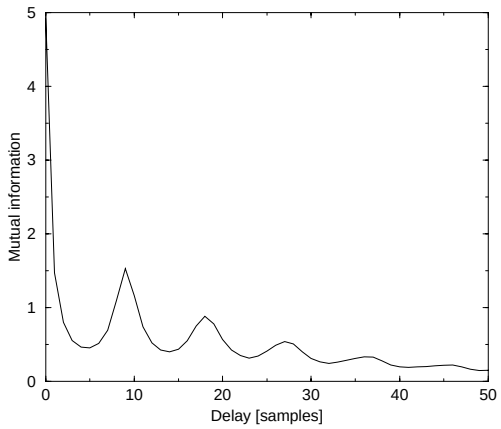


Figure 5.31: *Mutual information plot of the recursively predicted time series which was produced using a NRBFNP-RSC with an embedding dimension of 10, 500 kernels, an embedding delay of 8 samples, a prediction step of 1 sample, and a training length of 4000 samples. 25600 samples were used to estimate the mutual information.*

As can be seen from Figures 5.31, and 5.23(d), there is a significant difference between the mutual information plots of the recursively predicted and actual Dawber in-phase time series, respectively. This suggests that the actual clutter and recursively predicted data sets have different properties (*i.e.* different dynamics). Also note the difference in the correlation dimension estimate for the recursively predicted time series, with that given for the actual time series in Table 5.4. The difference in the mutual information plots for the original, and recursively predicted time series, would suggest that these two signals are significantly different. Furthermore the difference in correlation dimension might be taken as evidence that the recursively predicted time series does not share the same dynamics as the actual time series, if indeed the actual time series *is* chaotic and *has* associated dynamics. Before coming to a conclusion on the evidence of the underlying dynamics of the Dawber sea clutter data sets, it was decided that the other Dawber data sets would also be analysed for evidence of underlying dynamics.

The NMSE training, testing, and validation errors for the NRBFNP-RSC with an embedding dimension of 10, 500 kernels, an embedding delay of 8 samples, a prediction step of 1 sample, and a training length of 4000 samples, were -19.701821dB, -12.945741dB, and -10.630493dB, respectively. These values support the observation already made in section 5.5.4, that the predictor function for the Dawber quadrature channel data sets is linear: see the LP results in Figure 5.25.

Recursive prediction was attempted using NRBFNP-RSC's with embedding dimensions of 10, 14, and 14 for the Dawber VV amplitude, the Dawber HH in-phase and the Dawber HH amplitude data sets, respectively. For the recursive prediction of each data set the following NRBFNP-RSC parameters were used: 500 kernels, a prediction step of 1 sample, and embedding delays of 1 sample, 2 samples,..., 10 samples. For each different embedding delay, simulations were run for training lengths of 1000 samples up to 10,000 samples, in steps of 1000 samples.

Recursive prediction was successful for certain embedding delays and training lengths for the Dawber VV amplitude and Dawber HH in-phase data sets. However, as with the initial Dawber VV in-phase recursive prediction results shown in Figure 5.29, the recursively predicted data was not a very good representation of the actual time series. No successful recursive prediction results were obtained for the Dawber HH amplitude data set.

To summarise, the recursive prediction results for the Dawber amplitude and quadrature channel

data sets would seem to suggest that the Dawber data sets do not have any underlying dynamics. The reasons for this assertion are now discussed. For all the data sets analysed using recursive prediction, a successful, reasonably fair representation of the original signal was not achieved using Takens' embedding delay theorem. Takens' theorem is a known sufficient condition for dynamical reconstruction of chaotic signals (assuming a smooth observation map), and it was shown to be sufficient for the recursive prediction of Lorenz data, even when it was embedded in noise. Furthermore, in only 1 out of 4 data sets analysed was there achieved a reasonably fair recursively predicted representation of the original signal, and this was shown to have different properties to the original signal. Moreover, for the recursively predicted signal, given in Figure 5.30, there was no evidence of a prediction horizon, which is a characteristic of chaotic signals. What these results seem to point to, is that the underlying dynamics could not be captured, because there were no underlying dynamics to capture. This would support the conclusions reached in section 5.5.5, about the linear stochastic nature of the Dawber sea clutter data sets.

5.6 Summary of results for the Dawber data sets

The prediction results presented in sections 5.5.2 and 5.5.4, for the Dawber HH and VV amplitude and quadrature channel data sets suggest that these data sets have a linear predictor function. This evidence, combined with the lack of evidence in section 5.5.6 for the existence of underlying dynamics, would suggest that the Dawber sea clutter data sets are not chaotic. Indeed, evidence was presented in section 5.5.5 which suggested that a compound Gaussian stochastic model was more appropriate.

A possible explanation for the lack of evidence of underlying dynamics might be that the range resolution of the radar used to collect the Dawber data sets was too low. The range resolution of the radar was 1μs. However, in [21] Haykin and Puthusserypady found evidence of chaotic dynamics with data that used a radar with a range resolution of 2μs. Therefore, the resolution was not considered to be a reason for the lack of evidence of chaotic dynamics. Besides, if *all*[7] clutter is a chaotic process, this should be evident not only at high but also at low resolutions.

Another reason for the lack of evidence of underlying nonlinear dynamics and chaotic behaviour could be due to the fact that the Dawber data records represent only a very short time

[7] *i.e.* that there exists a nonlinear deterministic mechanism responsible for the generation of all clutter, which is what is claimed by Haykin and Puthusserypady.

period, spanning only 1.28 seconds. These clutter data record lengths are much shorter than those analysed, for example, by Haykin and Puthusserypady [21] for evidence of chaotic behaviour, who used data records which lasted for between 25 and 50 seconds. In other words, the Dawber data sets may appear to be linear over short time periods, such as 1.28 seconds, but over longer time periods nonlinear, chaotic behaviour might be detected. This discussion is similar to that given in section 4.1, about the use of locally linear predictors by Casdagli, to predict Lorenz data. From a practical perspective, it is significant that chaotic behaviour cannot be detected during a period of 1.28 seconds. For example, in a scanning radar the dwell time is typically on the order of a few milliseconds, and if chaotic behaviour is not observable during periods on the order of a second, but only over longer time periods, then chaotic behaviour could not be exploited to improve the performance of scanning radars. In other words, the debate over whether or not clutter is chaotic would be irrelevant for scanning radars.

5.7 Chapter summary

A comprehensive prediction analysis of the wavetank and Dawber sea clutter data sets was presented in this chapter. These clutter data sets were collected using a land-based radar that was operated in a dwelling mode, that is, with the antenna pointing towards a patch of sea surface along a fixed direction. The prediction analysis of the wavetank and Dawber data sets was carried out using global linear and nonlinear predictors on the in-phase, quadrature-phase, and amplitude channels of the radar data sets. In particular the nonlinear prediction analysis was carried out using RBFNP's with a wide range of embedding dimensions, number of kernels, embedding delays, and prediction steps. This nonlinear prediction analysis included using embedding parameters which adhered to Takens' embedding theorem for dynamical reconstruction. Results were presented which showed that the wavetank and Dawber data sets have linear predictor functions. Additionally, recursive prediction was carried out on these clutter data sets which demonstrated a lack of evidence of any underlying nonlinear dynamics. These results suggest that the wavetank and Dawber data sets are not examples of chaotic time series. Indeed, evidence was presented which showed that a compound Gaussian stochastic model is appropriate for the wavetank and Dawber clutter data sets. Furthermore, these results are in contrast to those published during the last 10 years by Simon Haykin and his co-workers, who have argued and provided evidence to support the following:

- that sea clutter has a nonlinear predictor function,
- that the underlying dynamics of sea clutter can be modelled using a nonlinear network (in recursive prediction mode), and
- that sea clutter is a chaotic process.

The data sets used to come to these conclusions were collected using a land-based radar operated in a dwelling mode. Despite collecting data using the same radar mode, it was pointed out that there were differences, in terms of radar and environmental parameters, between Haykin and his co-workers' clutter data sets, and the clutter data sets described in Appendix A. It was stated that these differences might be responsible for the lack of evidence of chaotic behaviour. In the case of the wavetank data sets, chaotic behaviour may have been concealed due to pulse compression. In the case of the Dawber data sets, chaotic behaviour may not have been noticed because the data sets analysed were collected during a much shorter time period than those analysed by Haykin and his co-workers, and they could just represent a locally linear section of an overall nonlinear process.

Chapter 6
Nonlinear prediction of sea clutter: further analysis

6.1 Introduction

The purpose of this chapter is to investigate if there are nonlinearities in the sea clutter data sets, which are described in Appendix A, that were not identified in the prediction analysis reported in Chapter 5. This is done as follows. In section 6.2 and 6.3, filters are applied to the clutter data sets, in order to try and reduce noise effects such as receiver and quantisation noise. Prediction analysis is then carried out on the filtered time series. In section 6.2 a smoothing filter is applied to the clutter, and in section 6.3 a linear phase FIR lowpass filter is used. Haykin and Puthusserypady [21] used these filtering techniques when calculating the chaotic invariants of their clutter data, however, they did not carry out a prediction analysis on their filtered clutter. In section 6.4 a novel structure for a forward-backward [116] RBFNP is presented, and forward-backward linear and nonlinear prediction of sea clutter is reported. A chapter summary is presented in section 6.5.

6.2 Applying a Smoothing filter to the clutter data before prediction analysis

In their analysis of sea clutter [21], Haykin and Puthusserypady used the simple smoothing filter, given in equation (6.1), to reduce the effect of experimental noise, and to obtain the smoothed clutter sequence $\{x_s(n)\}$ from the original clutter sequence $\{x(n)\}$.

$$x_s = \sum_{i=1}^{3} \frac{1}{3} x(n+i-2) \tag{6.1}$$

This smoothing filter was applied to the Dawber VV amplitude and in-phase channel data sets. The PSD plots for the un-smoothed and smoothed data sets are given in Figure 6.1.

The mutual information plots for the smoothed data sets are given in Figure 6.2. It can be seen there are no significant differences between these plots and those for the un-smoothed data sets, given in Figures 5.19(d) and 5.23(d) for the amplitude and in-phase data sets, respectively.

Correlation dimension estimates for the smoothed data sets are presented in Table 6.1. For convenience, the estimates already presented for the un-smoothed Dawber data sets are repeated again in Table 6.1.

Data set	**Correlation dimension**
VV amplitude	3.45714
VV amplitude smoothed	3.91139
VV in-phase	3.47368
VV in-phase smoothed	4.39024

Table 6.1: *Maximum likelihood correlation dimension D_{ML} estimates for the smoothed Dawber sea clutter data sets.*

Initial prediction results for the smoothed data sets are given in Figure 6.3. An embedding delay of 1 sample was used as for all previous analysis presented in this thesis it produced the best prediction performance, given all other parameters were constant. The correlation dimension estimates in Table 6.1 were used to select a suitable embedding dimension. As can be seen in Figure 6.3, LP results were always as good as, or better than the NLP results. In fact, the generalisation properties of the NRBFNP-RSC were much poorer than those of the LP. The generalisation properties of the NRBFNP-RSC for the un-smoothed Dawber VV amplitude data set, see Figure 5.20, were poorer than for the LP, and these poor generalisation properties appear to have been amplified after applying the smoothing filter.

A more rigorous prediction analysis was carried out for a variation of NRBFNP-RSC free parameters. These results are given in Figure 6.4. The results given in Figure 6.4 provide evidence to suggest that the smoothed data sets have a linear predictor function. The spurious large NMSE values in Figures 6.4(c) and 6.4(d) are a result of ill-conditioning. This matter was discussed in section 5.5.2.

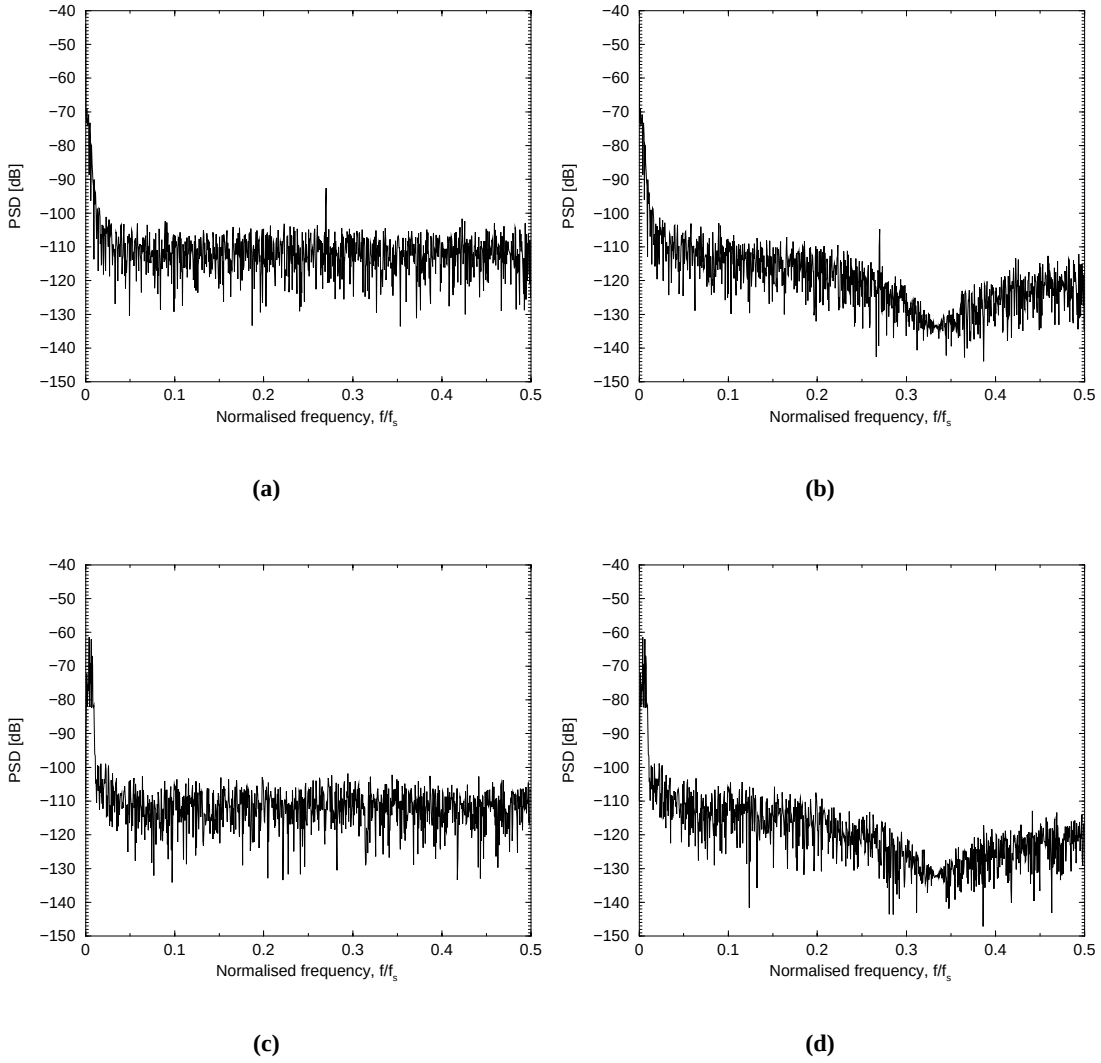
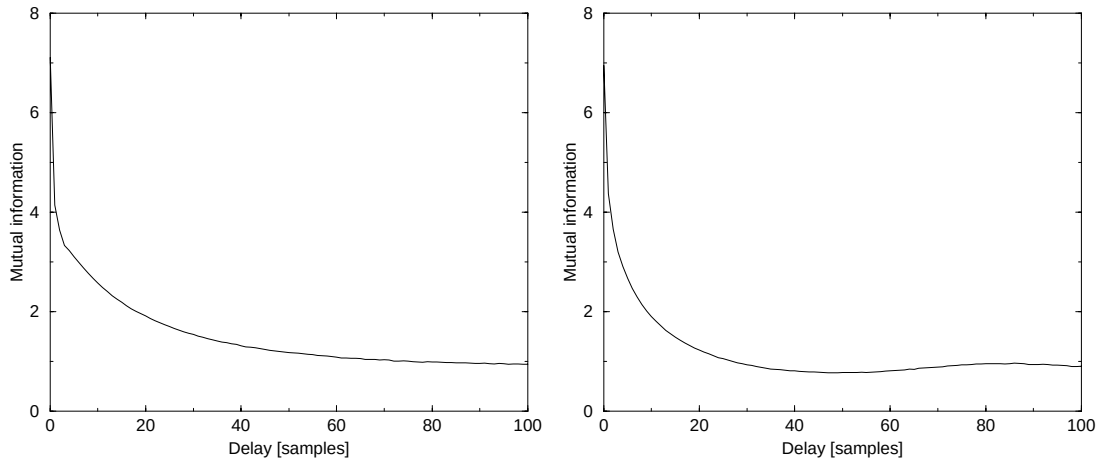
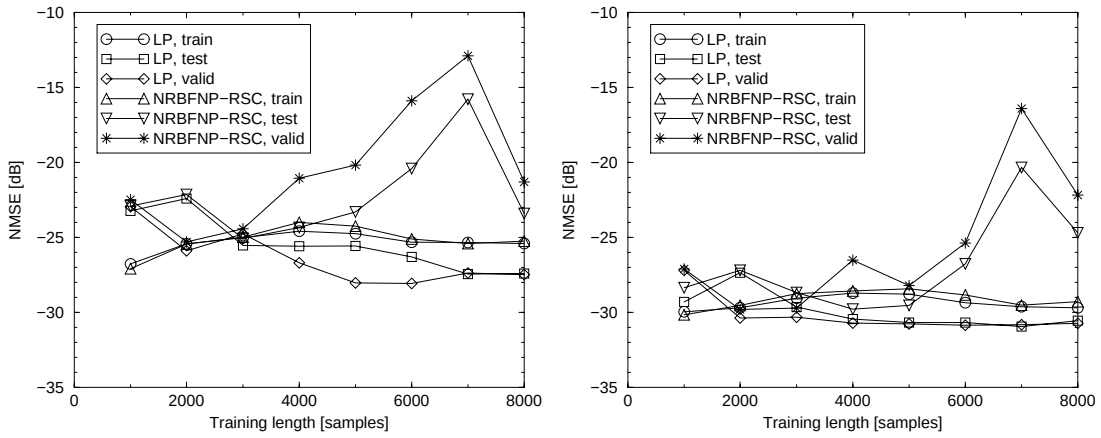


Figure 6.1: *PSD's for (a) the smoothed, and (b) the un-smoothed Dawber VV amplitude data, and for the (c) smoothed, and (d) the un-smoothed Dawber VV in-phase data.*



(a) **(b)**

Figure 6.2: *Mutual information plots for (a) the smoothed Dawber VV amplitude data, and (b) the smoothed Dawber VV in-phase data.*



(a) **(b)**

Figure 6.3: *Initial 1-step ahead prediction analysis of the (a) smoothed Dawber VV amplitude and (b) smoothed Dawber VV in-phase data sets, using a NRBFNP-RSC with an embedding dimension of 10, 100 kernels and an embedding delay of 1 sample. Results are also shown for a 10 tap LP with a delay of 1 sample between each tap.*

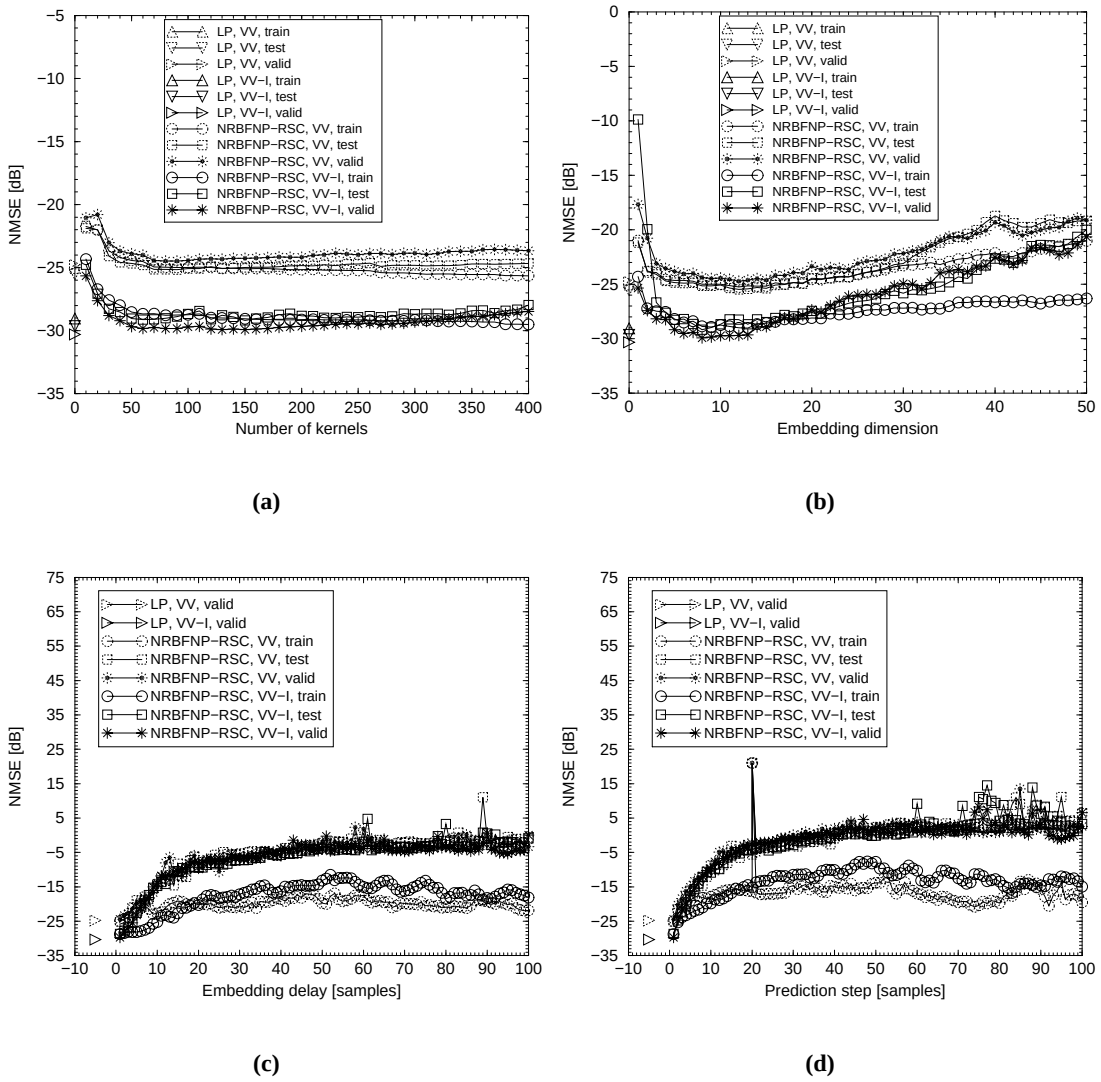


Figure 6.4: *Further prediction analysis of the smoothed Dawber VV amplitude (VV) and in-phase (VV-I) data sets, using NRBFNP-RSC's with a training length of 3000 samples: (a) NMSE vs number of kernels using an embedding dimension of 10, an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (b) NMSE vs embedding dimension using 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (c) NMSE vs embedding delay using an embedding dimension of 10, 100 kernels, and a prediction step of 1 sample for both data sets, (d) NMSE vs prediction step using an embedding dimension of 10, 100 kernels, and an embedding delay equal to the prediction step for both data sets. Linear 1-step ahead prediction results are also shown in each plot for a 10 tap LP (with a 1 sample delay between each tap input) which had a training length of 3000 samples.*

6.3 Applying a Linear phase finite impulse response lowpass filter to the clutter before prediction analysis

In addition to the smoothing filter discussed in section 6.2, Haykin and Puthusserypady [21] used a finite impulse response (FIR) [127] filter to suppress noise, in their analysis (from a chaos theory perspective) of sea clutter. Therefore, this type of filter was also used, to see if it could suppress noise and improve clutter predictability, and if it could be used to uncover any hidden nonlinearities. Haykin and Puthusserypady used a linear phase FIR filter with 100 taps, and a low frequency passband which was chosen in an attempt to suppress noise, but which would avoid altering the spectra of the actual clutter data. For the analysis in this thesis, the Dawber VV amplitude and in-phase data sets were filtered using a lowpass linear phase FIR filter with 101 taps, and a cut-off frequency of 0.05 normalised frequency. The corresponding filtered PSD plots, along with the PSD plots for the un-filtered Dawber data sets are given in Figure 6.5.

The mutual information plots for the filtered data sets are given in Figure 6.6. It can be seen there are no significant differences between these plots, and those for the un-filtered data sets, given in Figures 5.19(d) and 5.23(d) for the amplitude and in-phase data sets, respectively.

Correlation dimension estimates for the filtered data sets are presented in Table 6.2. For convenience, the estimates already presented for the un-filtered Dawber data sets are repeated again in Table 6.2.

Data set	Correlation dimension
VV amplitude	3.45714
VV amplitude filtered	6.33334
VV in-phase	3.47368
VV in-phase filtered	4.64045

Table 6.2: *Maximum likelihood correlation dimension D_{ML} estimates for the filtered Dawber sea clutter data sets.*

Initial prediction results for the FIR filtered data sets are given in Figure 6.7. For the reason already discussed in section 6.2 an embedding delay of 1 sample was used. The correlation dimension estimates in Table 6.2 were used to select a suitable embedding dimension. As can be seen in Figure 6.7, LP results were always as good as, or better than the NLP results. In fact, the generalisation properties of the NRBFNP-RSC were much poorer than those for the LP.

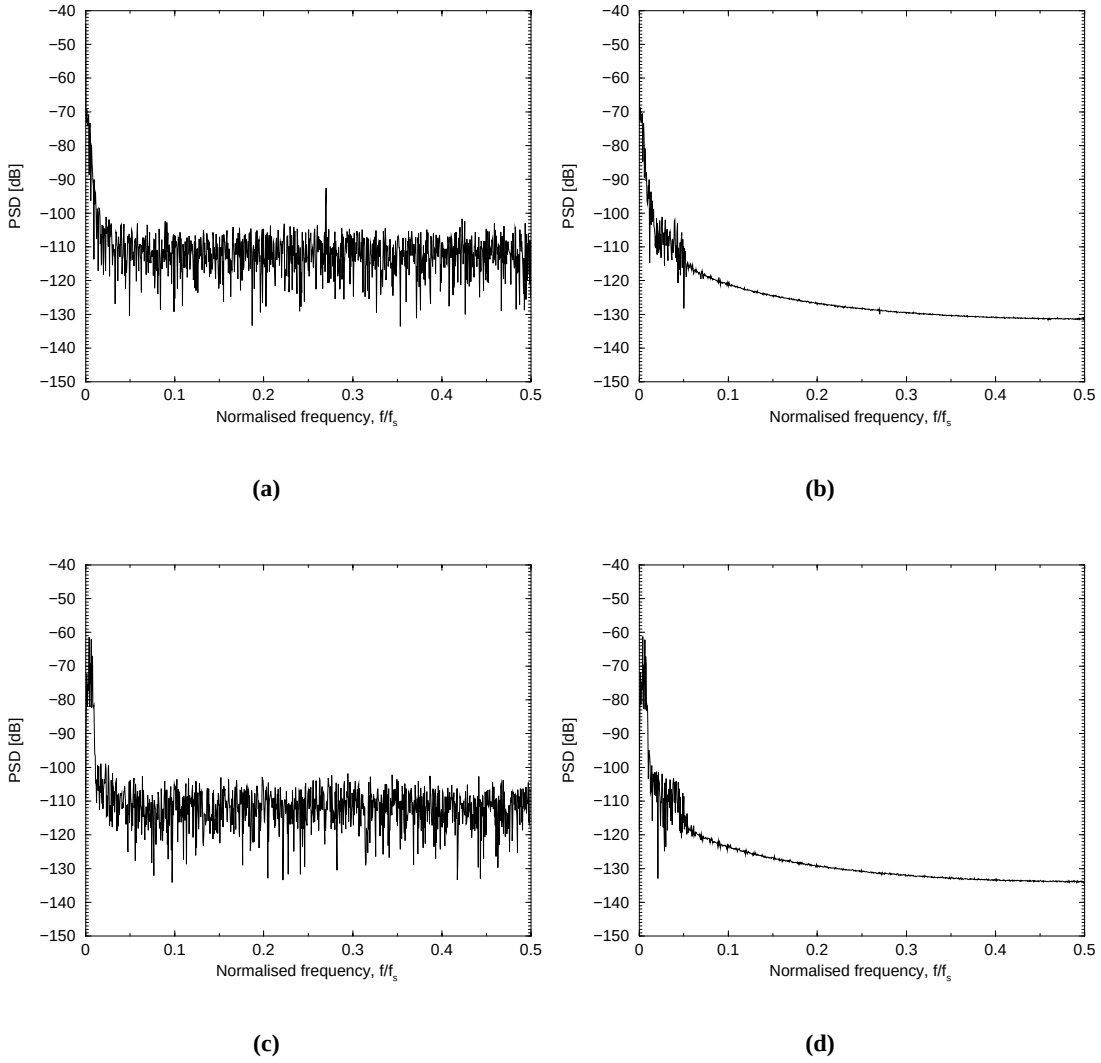
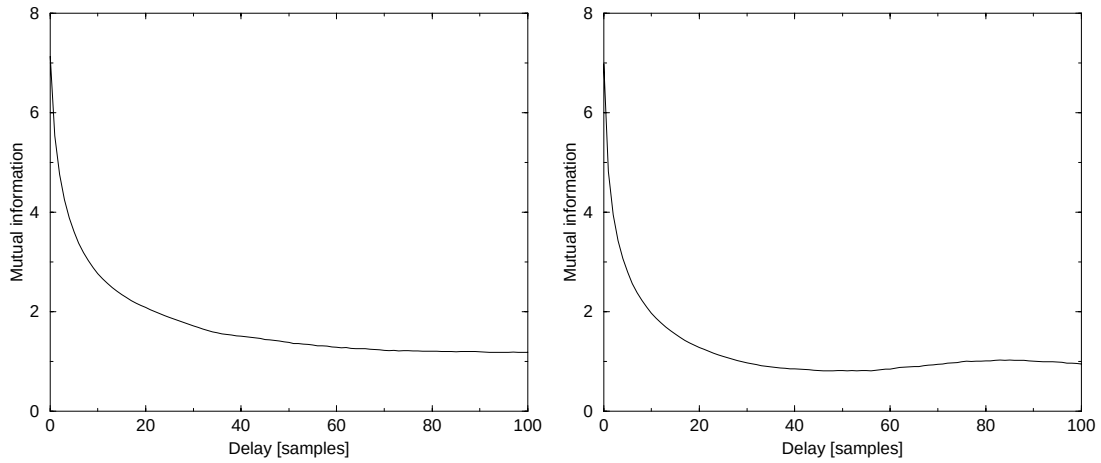
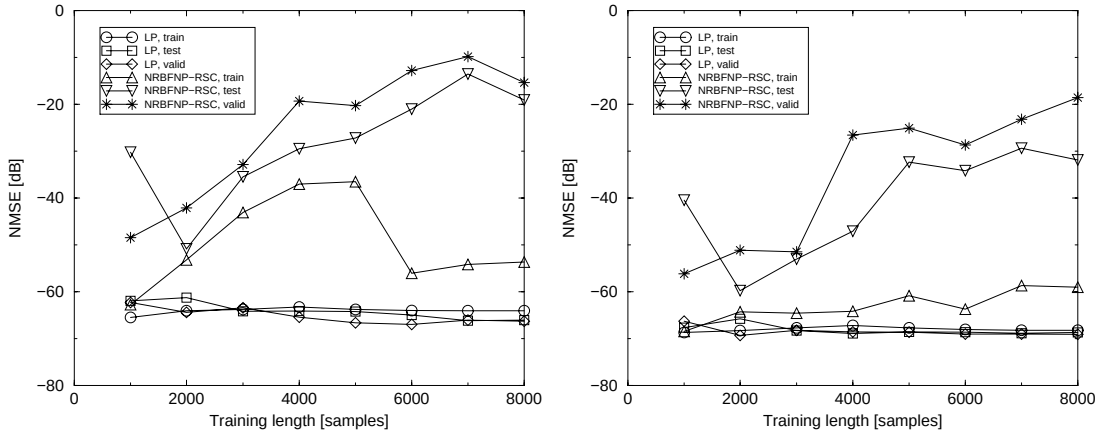


Figure 6.5: *PSD's for (a) the un-filtered, and (b) the FIR filtered Dawber VV amplitude data, and for the (c) un-filtered, and (d) the FIR filtered Dawber VV in-phase data.*



(a) **(b)**

Figure 6.6: *Mutual information plots for (a) the FIR filtered Dawber VV amplitude data, and (b) the FIR filtered Dawber VV in-phase data.*



(a) **(b)**

Figure 6.7: *Initial 1-step ahead prediction analysis of the (a) FIR filtered Dawber VV amplitude and (b) FIR filtered Dawber VV in-phase data sets, using a NRBFNP-RSC with an embedding dimension of 14 for the results in (a), and 12 for those in (b), with 100 kernels and an embedding delay of 1 sample used for the results in both (a) and (b). Results are also shown for a 10 tap LP.*

The generalisation properties of the NRBFNP-RSC for the un-filtered Dawber VV amplitude data set, see Figure 5.20, were poorer than for a LP, and these poor generalisation properties appear to have been amplified after applying the FIR filter, even more so, than for the case of the smoothing filter (see Figure 6.3). A more rigorous prediction analysis was carried out for a variation of the NRBFNP-RSC's free parameters. These results are given in Figure 6.8. The results given in Figure 6.8 provide further evidence to suggest that the filtered data sets have a linear predictor function. The large spurious NMSE values obtained in Figures 6.8(c) and 6.8(d) are due to ill-conditioning. This matter was discussed in section 5.5.2.

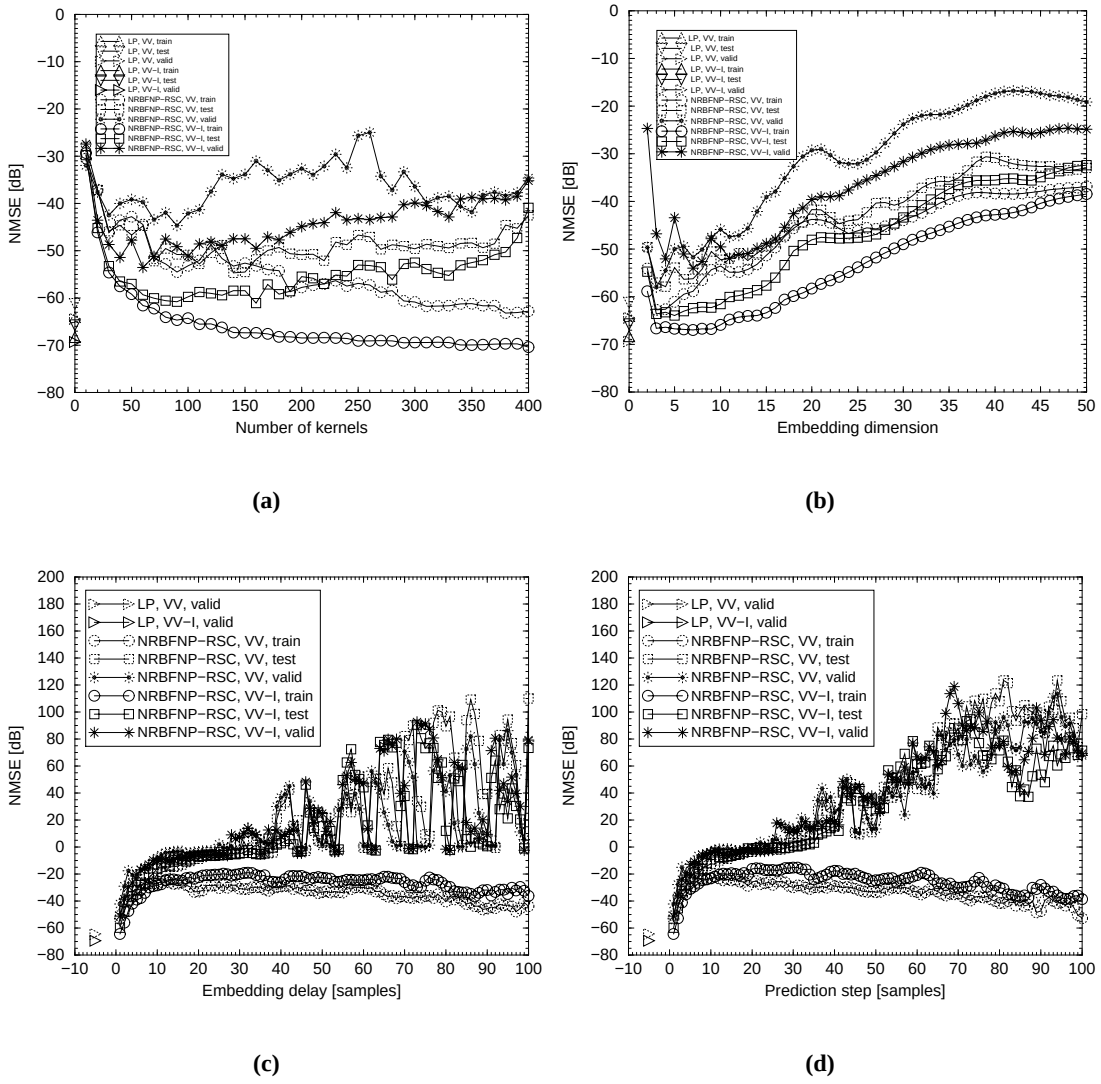


Figure 6.8: *Further prediction analysis of the FIR filtered Dawber VV amplitude (VV) and in-phase (VV-I) data sets, using NRBFNP-RSC's with a training length of 2000 samples: (a) NMSE vs number of kernels using an embedding dimension of 14 for the VV data and 12 for the VV-I data, with an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (b) NMSE vs embedding dimension using 100 kernels, an embedding delay of 1 sample, and a prediction step of 1 sample for both data sets, (c) NMSE vs embedding delay using an embedding dimension of 14 for the VV data and 12 for the VV-I data, with 100 kernels, and a prediction step of 1 sample for both data sets, (d) NMSE vs prediction step using an embedding dimension of 14 for the VV data and 12 for the VV-I data, with 100 kernels, and an embedding delay equal to the prediction step for both data sets. Linear 1-step ahead prediction results are also shown in each plot for a 10 tap LP (with a 1 sample delay between each tap input) which had a training length of 2000 samples.*

6.4 Forward-backward prediction

Until now, the emphasis in this thesis has been on forward prediction of data. Attention is now turned to forward-backward (FB) [116] prediction. It is well known that FB prediction can be used to improve prediction performance with respect to forward prediction alone. Of additional interest in applying the method of FB prediction to sea clutter, is to investigate whether nonlinearities not observable using forward prediction, are observable using FB prediction.

6.4.1 Forward-backward RBFN

The structure of the forward-backward RBFN (FB-RBFN) used for this investigation is given in Figure 6.9.

The FB-RBFN is shown with $2N$ input nodes: N input nodes either side of the test cell, which it tries to predict. The forward section of the FB-RBFN is associated with the input nodes, 1 through to N. The backward section is associated with the input nodes $N+1$ through to $2N$. In order to compare the prediction performance of a forward RBFN with that of a FB-RBFN, the centres used in the forward section were also used by the backward section. As with the forward RBFN case, centres could be selected using the RSC or OAKM methods. The FB-RBFN's $2M$ weights were trained using the Householder transform, in the same way that the M weights of the forward RBFN were trained. To the best of the author's knowledge, this structure for a FB-RBFN is novel, as is the training methodology.

6.4.2 Forward-backward prediction of Lorenz data

To demonstrate the improvement gained by the use of FB prediction, forward and FB linear and nonlinear prediction results for the noise-free Lorenz data, described in section 4.7.1, are shown in Figure 6.10.

As can be seen from Figure 6.10, FB prediction improved the performance of both the linear and nonlinear predictors. The FBLP performed approximately 30dB better than the LP, and the FB-NRBFNP-RSC managed to perform more than 20dB better than the NRBFNP-RSC, for a training length of 10,000 samples. Interestingly, the FBLP managed to achieve a similar performance to that of the NRBFNP-RSC.

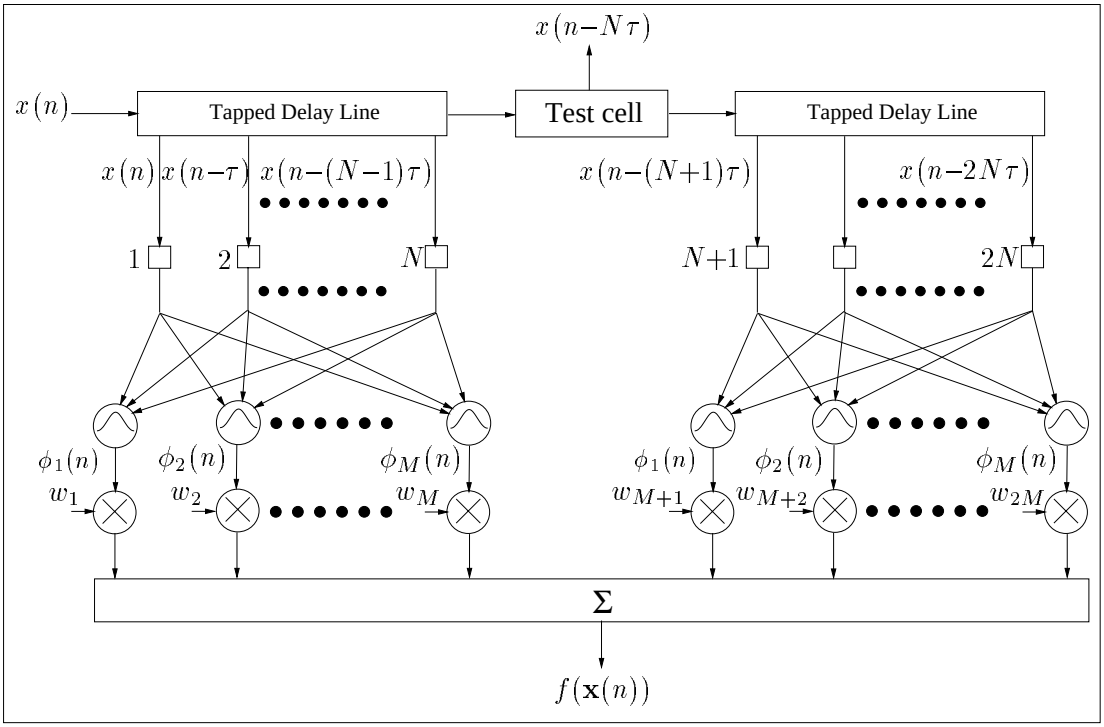
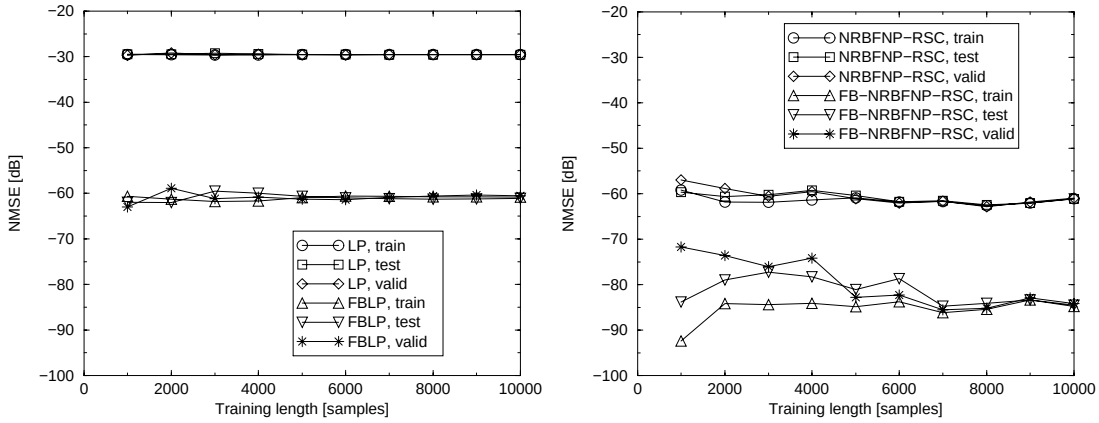


Figure 6.9: *Forward-backward RBFN.*

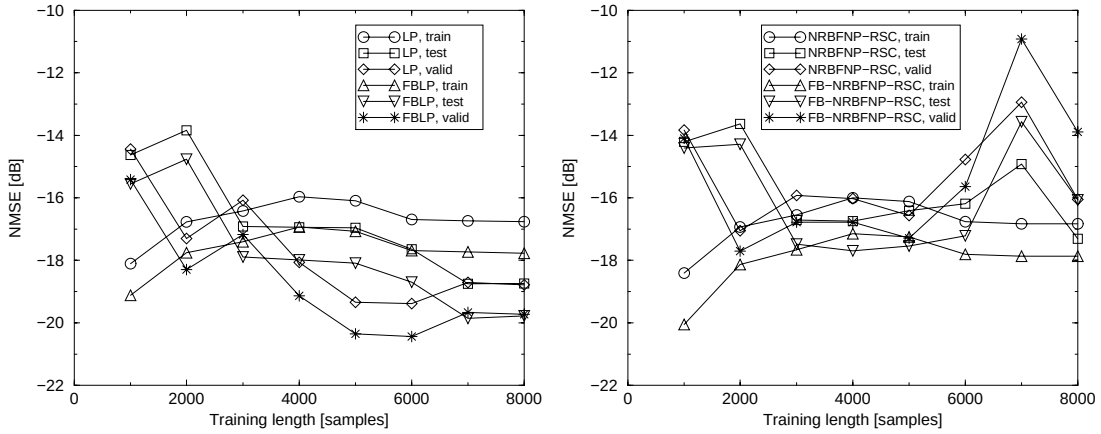


(a) **(b)**

Figure 6.10: *Forward-backward (a) linear and (b) nonlinear prediction of noise-free Lorenz data. Results are shown in (a) for a (forward) LP which used 20 taps, and for a FBLP which used 20 taps in the forward section and 20 taps in the backward section. Results are shown in (b) for a NRBFNP-RSC which used an embedding dimension of 3, 100 kernels and an embedding delay of 1 sample, and for a FB-NRBFNP-RSC which used an embedding dimension of 3, 100 kernels and an embedding delay of 1 sample in both the forward and backward sections.*

6.4.3 Forward-backward prediction of Dawber VV amplitude data

Forward-backward prediction results are given in Figure 6.11, for the Dawber VV amplitude data set.



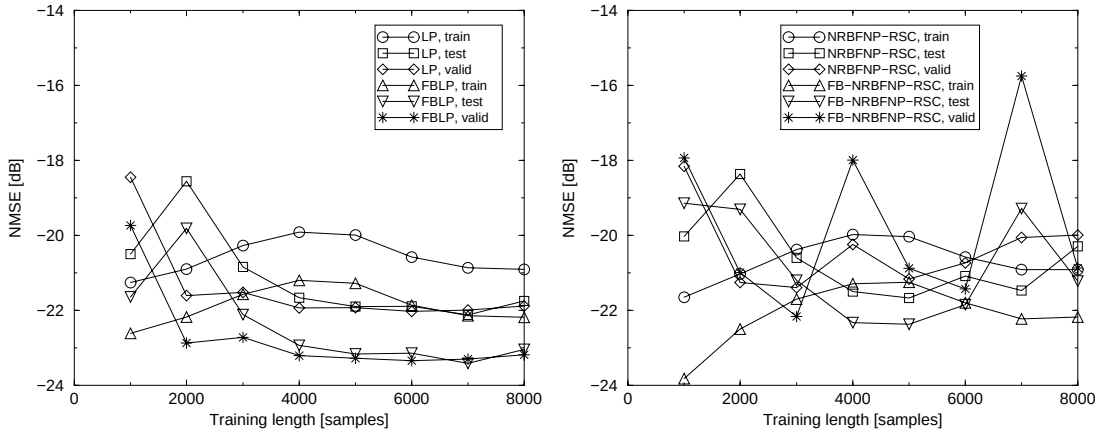
(a) **(b)**

Figure 6.11: *Forward-backward (a) linear and (b) nonlinear prediction of Dawber VV amplitude data. Results are shown in (a) for a (forward) LP which used 10 taps, and for a FBLP which used 10 taps in the forward section and 10 taps in the backward section. Results are shown in (b) for a NRBFNP-RSC which used an embedding dimension of 10, 100 kernels and an embedding delay of 1 sample, and for a FB-NRBFNP-RSC which used an embedding dimension of 10, 100 kernels and an embedding delay of 1 sample in both the forward and backward sections.*

As can be seen from Figure 6.11, the FB predictors did manage to achieve slightly better prediction results than the forward predictors. The improvement margin was approximately 1 dB. Note however, that in the case of the FB-NRBFNP-RSC, after a training length of 5000 samples, this predictor started to over-fit to the data, and as a consequence its generalisation properties became poorer. Similarly poor NRBFNP-RSC generalisation properties were observed for the (forward) prediction of the VV amplitude, smoothed VV amplitude and FIR filtered VV amplitude data sets. The similar performance of the FBLP and of the FB-NRBFNP-RSC on training data, coupled with the fact that the FBLP had better generalisation properties adds further evidence to suggest that this data has a linear predictor function.

6.4.4 Forward-backward prediction of Dawber VV in-phase channel data

Forward-backward prediction results are given in Figure 6.12, for the Dawber VV in-phase channel data set. As can be seen from Figure 6.12, the forward-backward predictors did man-



(a) **(b)**

Figure 6.12: *Forward-backward (a) linear and (b) nonlinear prediction of Dawber VV in-phase data. Results are shown in (a) for a (forward) LP which used 10 taps, and for a FBLP which used 10 taps in the forward section and 10 taps in the backward section. Results are shown in (b) for a NRBFNP-RSC which used an embedding dimension of 10, 100 kernels and an embedding delay of 1 sample, and for a FB-NRBFNP-RSC which used an embedding dimension of 10, 100 kernels and an embedding delay of 1 sample in both the forward and backward sections.*

age to achieve slightly better prediction results than the forward predictors. The improvement margin was approximately 1 dB for the FBLP, with respect to the LP. The FBLP and the FB-NRBFNP-RSC performed similarly on training data, however the FBLP performed better on non-training data sets. The similar performance of the FBLP and of the FB-NRBFNP-RSC on training data, coupled with the fact that the FBLP had better generalisation properties adds further evidence to suggest that this data has a linear predictor function.

6.5 Chapter summary

To determine if receiver and/or measurement noise were responsible for the lack of evidence of chaotic behaviour, the Dawber clutter data sets were first filtered and then a prediction analysis was carried out. This filtering was performed using two techniques employed by Haykin and

Puthusserypady [21] in their measurement of chaotic invariants from sea clutter data sets. The first technique was a smoothing filter, which simply produced an average of 3 clutter samples at a time. The second was a FIR lowpass filter. The prediction analysis of the filtered data was carried out using global linear and nonlinear predictors. As was the case for the analysis reported in Chapter 5, RBFNP's were used with a wide range of embedding parameters, number of kernels, and prediction steps. The results of this analysis provided further evidence to suggest that the Dawber clutter data sets have linear predictor functions, and are consequently not chaotic. In other words, noise was not thought to be responsible for the concealment of chaotic behaviour.

A novel structure for a forward-backward RBFNP was proposed. Forward-backward linear and nonlinear (global) prediction was carried out on the Dawber data sets in order to establish if there was any evidence of nonlinear predictability that was not detected with forward prediction analysis alone. The results obtained showed that there was no evidence of nonlinear predictability using forward-backward prediction.

The results presented in this chapter provide further evidence to that already given in Chapter 5, which suggested that the clutter data sets analysed in this thesis have linear predictor functions, and are consequently not examples of chaotic time series.

Chapter 7
Radar target detection

7.1 Introduction

As reported in Chapter 5, a comprehensive prediction analysis was carried out for the sea clutter data sets described in Appendix A. The most significant conclusion reached from that analysis was that the clutter data sets were found to have linear predictor functions: LP NMSE results were found to be as good as, or better than NLP NMSE results. Furthermore, evidence was presented in Chapter 5, which suggested that the clutter data sets analysed are best modelled as a stochastic, locally Rayleigh, process and not, as is claimed in [9–21], as a deterministic chaotic process. The linear predictability of the clutter data sets was attributed to Rayleigh (speckle) correlations. In [12] evidence was presented which showed that a NLPD could outperform both a LPD [128], and a "standard" non-coherent detector. A similar comparison was carried out in [17], which compared the performance of a NLPD and a conventional radar (CFAR) detector. This study showed that a NLPD could perform better than a conventional CFAR detector. The implications of the prediction results obtained in Chapter 5, for single pulse radar target detection in a marine environment, are discussed in this chapter as follows. In section 7.2 some basic radar target detection concepts are presented. A discussion of radar detection techniques used in this chapter is given in section 7.3. Before presenting any detection results using clutter data, Lorenz data (described in section 4.7.1) is used to assess which is more important for better target detection results using a predictor-detector: is it NMSE, or is it capturing a signal's underlying dynamics (assuming that the signal is chaotic)? This question is investigated in section 7.4. In section 7.5, results for single pulse radar target detection, against a sea clutter background are presented. A chapter summary is presented in section 7.6.

7.2 Basic radar target detection concepts

7.2.1 Introduction

Maritime surveillance radar is used to detect targets in an ocean environment, where the radar reflections from the sea surface, the sea clutter, usually are the main sources of interference, which degrade the performance of the radar: the clutter to thermal noise ratio is often large enough for the thermal noise to be ignored in the design and performance evaluation of a maritime surveillance radar. The term *target* in a marine environment is used to refer to any radar backscatter that does not come from the ocean surface. Such backscatter may come from man-made objects (*i.e.* a submarine periscope or a ship), or some natural object (*i.e.* an iceberg).

7.2.2 Radar target detection: elementary definitions and jargon

Given a backscattered sample which could consist entirely of reflections from the sea, or which could be composed of target reflections as well as reflections from the sea, there are two possible options: declare that no target is present, or alternatively, declare that a target is present in the received sample. For each option, there are two consequences:

- if "no target" is opted for, and a target is present in the recovered sample, the consequence is a missed target detection.
- if "no target" is opted for, and no target is present in the received sample, the consequence is a successful non-detection.
- if "target present" is opted for, but none is present, the consequence is referred to as a false alarm.
- if "target present" is opted for, and a target is present, the consequence is a successful target detection.

To distinguish between *clutter* and *clutter plus target* returns, a threshold can be used: any returns above this threshold will be classified as "clutter plus target", and any below, as "clutter only". Such an approach has an obvious trade-off: setting the threshold at very low values will improve the probability of successfully detecting targets, but will equally increase the number of false alarms, as more clutter samples will cross the threshold. This trade-off is illustrated in Figure 7.1.

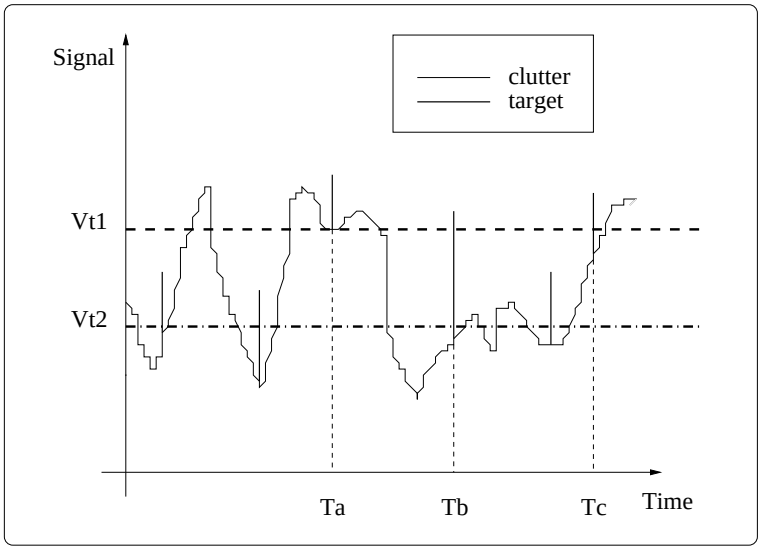


Figure 7.1: *Threshold tradeoff. Having the threshold level set at Vt1 results in a low false alarm count (*i.e. *the number of times the clutter crosses the threshold). However, only the target samples at times Ta, Tb, and Tc are correctly detected. Decreasing the threshold level to Vt2 improves the probability of detection, but increases the probability of false alarm: more clutter samples now cross the threshold level.*

Two quantities used to describe the performance of a radar target detector are the probability of false alarm P_{fa} and the probability of detection P_d. To define these two quantities for sea clutter returns, consider that the clutter returns $\{x(n)\}$ are described by a pdf $P_x(x)$. Then the probability that a false alarm will occur (*i.e.* P_{fa}), is the probability that a clutter sample $x(n)$ will exceed the threshold of value, say, v_t, *i.e.*

$$P_{fa} = \int_{v_t}^{\infty} P_x(x)dx \tag{7.1}$$

Given that the pdf for an arbitrary target signal $t(n)$ is given by $P_t(t)$, and the signal plus clutter pdf is given by $P_{x+t}(x+t)$, then the probability that a signal will be detected (*i.e.* P_d), is the probability that a clutter plus signal sample $x(n)+t(n)$ will exceed the threshold v_t, *i.e.*

$$P_d = \int_{v_t}^{\infty} P_{(x+t)}(x+t)d(x+t) \tag{7.2}$$

Ideally, a radar target detector would always detect a target, when a target was present (i.e P_d=1), and it would never mistake clutter or noise, or any other interfering signal for a target, when no target was present, (*i.e.* P_{fa}=0). However, this is never achieved in practice, but the aim is to try and build a detector with a P_d as close to 1, and a P_{fa} as close to 0 as possible, for a given threshold value. One method for comparing the performance of radar target detectors is what is known as a receiver operating curve (ROC). This is simply a plot of P_{fa} versus P_d. ROC's are the chosen method for assessing detection performance in this thesis. ROC's were also used in [12, 17] to assess the detection performance of of a NLPD relative to a CFAR detector.

7.2.3 Target models

To assess the detection performance of a radar target detector, and therefore to plot a ROC, it is necessary to use some kind of target signal. It is often impractical to obtain experimental target data, and therefore a common approach within the radar community is to use what are referred to as Swerling target models [7, 126].

A Swerling case I, or simply a Swerling I, model has an intensity I_t, (*i.e.* power) pdf given by the exponential distribution, *i.e.*

$$P(I_t|\sigma_t) = \frac{1}{\sigma_t} \exp\left[\frac{-I_t}{\sigma_t}\right] \tag{7.3}$$

while the corresponding amplitude returns a_t are Rayleigh distributed,

$$P(a_t|\sigma_t) = \frac{a_t}{\sigma_t} \exp\left[\frac{-a_t^2}{2\sigma_t}\right] \tag{7.4}$$

where σ_t is the average power.

A Swerling I target fluctuates slowly, on a scan to scan basis. A Swerling II target is described by the same intensity pdf, equation (7.3), and the same amplitude pdf, equation (7.4), as the Swerling I target. However, the Swerling II target is modelled as fluctuating more quickly than

the Swerling I case: on a pulse to pulse basis, rather than on a scan to scan basis.

A Swerling III target has an intensity pdf given by the chi-square distribution, with four degrees of freedom, *i.e.*

$$P(I_t|\sigma_t) = \frac{4I}{\sigma_t^2} \exp\left[\frac{-2I_t}{\sigma_t}\right] \tag{7.5}$$

A Swerling III target fluctuates slowly, on a scan to scan basis. A Swerling IV target is described by the same intensity pdf as the Swerling III target, but is modelled as fluctuating more quickly, on a pulse to pulse basis.

A non-fluctuating (*i.e.* constant amplitude) target is often referred to as a Swerling 0 target.

The Swerling target models are summarised in Table 7.1.

Swerling target model	**Description**
0	Constant amplitude
I	Rayleigh amplitude pdf, slowly fluctuates
II	Rayleigh amplitude pdf, quickly fluctuates
III	Chi-square intensity pdf, slowly fluctuates
IV	Chi-square intensity pdf, quickly fluctuates

Table 7.1: *Swerling target models.*

Swerling I and II target models represent targets which consist of a large number of independent scatterers, with no one scatterer contributing more than a small fraction of the total backscattered energy. Swerling III and IV target models are indicative of scattering from one large dominant scatterer together with a collection of small independent scatterers.

It has been suggested that if only one parameter is used to describe a complex target, it should be the median value of the target amplitude returns, with Rayleigh statistics (*i.e.* Swerling I and II models) [7]. Furthermore, once it has been decided to use a Rayleigh target model, quite often the Swerling I model is used to assess radar performance in preference to the Swerling II model, since the Swerling I model results in a more conservative performance evaluation [7].

For the purposes of this thesis, the signal to clutter ratio (SCR) is defined as in equation(7.6),

$$\mathrm{SCR} = 10 \log_{10} \left[\frac{\sigma^2_{signal}}{\sigma^2_{clutter}} \right] \tag{7.6}$$

where σ^2_{signal} is the variance of the target signal, and $\sigma^2_{clutter}$ is the variance of the clutter signal.

7.3 Detection techniques

7.3.1 Introduction

This section discusses the detection techniques used in this thesis. Firstly, detectors based on linear and nonlinear predictors are discussed in section 7.3.2. Then, standard constant false alarm rate (CFAR), and fixed threshold detectors are described in sections 7.3.3 and 7.3.4, respectively.

7.3.2 Predictor-detectors

The incorporation of a (forward) predictor into a radar detector is now discussed. A block diagram of a predictor-detector is given in Figure 7.2.

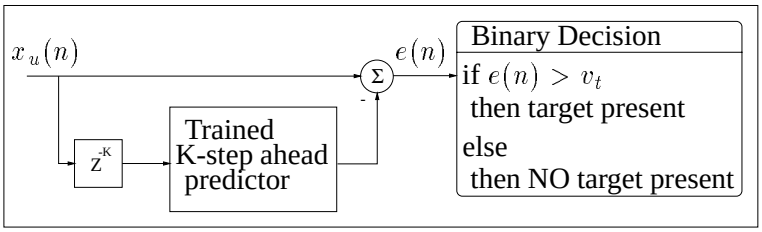


Figure 7.2: *Predictor-detector. A K-step ahead predictor (linear, VSF, or NRBFN) is trained as described in Chapter 4. Once trained, clutter samples which were not used to train the predictor $x_u(n)$ are presented to the predictor through a K-step delay. The clutter data $\{x_u(n)\}$ could consist of clutter alone, or clutter plus target. To determine if a target is present or not, the predictor error $e(n)$ is compared with a threshold level v_t. If the predictor error is greater than the threshold a target is declared to be present, and if it is less than the threshold level a target is not declared to be present.*

Note that, as depicted in Figure 7.2, detection analysis is carried out for a predictor-detector by thresholding the predictor error samples, $e(n)$. The concepts of section 7.2 can be extended to

the predictor error samples. For example, if an error sample crosses the error threshold (which thus results in the predictor-detector declaring a target is present), but no target is present in the sample being predicted, this is known as a false alarm.

For the analysis in this chapter, linear, VSF, and NRBFN predictors were used as predictor-detectors. Note that for all the NRBFNP-detector simulations discussed in this chapter, the RSC method was used to obtain centre positions for the NRBFNP, as there was found to be very little difference in the performance between the RSC and OAKM centre selection techniques, as discussed in Chapters 4 and 5. The RSC method was preferred as it is computationally more efficient. To obtain ROC plots for the predictor-detectors, the following approach was adopted.

Detection analysis using a predictor-detector:

- The predictor was trained, as discussed in Chapter 4, using a training data set, of length Y samples. The predictor errors $e(n)$ were collected during training.
- The next block of X samples, immediately following the training data set, were selected as a non-training data set which was used for the detection analysis, and to produce a ROC. The trained predictor was used on the non-training data set, and the predictor errors were collected.
- A desired SCR was chosen.
- The selected SCR was used to generate X Rayleigh variates with the appropriate variance. Note that the same target data set was used in each predictor-detector (LP, VSFP, NRBFNP) simulation, so that the performance of each predictor-detector could be compared.
- The absolute values of the errors (*i.e.* $|e(n)|$) collected during the training of the predictor were used to determine a suitable range of values for the threshold level, over which to carry out the detection analysis on the non-training data set. This was done as follows. The number of different v_t values to use was selected and this will be referred to as YY. The maximum and minimum absolute training error values were determined, and these will be referred to as e_{max} and e_{min} respectively. The range of v_t values used was, $\{e_{min}, \left[e_{min} + \left(\frac{e_{max}-e_{min}}{YY}\right)\right], \ldots, \left[e_{min} + (YY-1)\left(\frac{e_{max}-e_{min}}{YY}\right)\right]\}$.
- A Swerling I target was implemented as follows. A Rayleigh target sample was selected, from the X generated, as discussed above. This was added to each non-training data set

predictor error: in effect the Rayleigh sample was held constant for the duration of the non-training data record, X.

- P_{fa} estimates for a given v_t value were obtained by dividing the number of times the absolute predictor errors from the non-training data set crossed v_t, by the total number of non-training data set predictor errors.

- P_d estimates were obtained for a given v_t value and target sample $t(n)$ by dividing the number of times a non-training absolute predictor error plus target sample (*i.e.* $|e(n)| + t(n)$) crossed v_t, by the total number of non-training predictor errors. To obtain an average P_d estimate, for any v_t setting, the P_d estimates for each of the X different target samples were averaged.

- Using the estimated P_d and P_{fa} values, a ROC could be plotted for the predictor-detector.

7.3.3 Cell-averaging constant false alarm rate detector

A standard engineering approach to radar target detection is to design a detector which can maintain a constant false alarm rate. These detectors are known as constant false alarm rate (CFAR) detectors [6, 7, 126, 129, 130]. This is a broad class of radar detector, with many variants [7, 126, 131, 132]. The simplest type of CFAR detector is the cell-averaging CFAR (CA-CFAR) [7, 126, 129, 130] detector. The CA-CFAR is depicted in Figure 7.3.

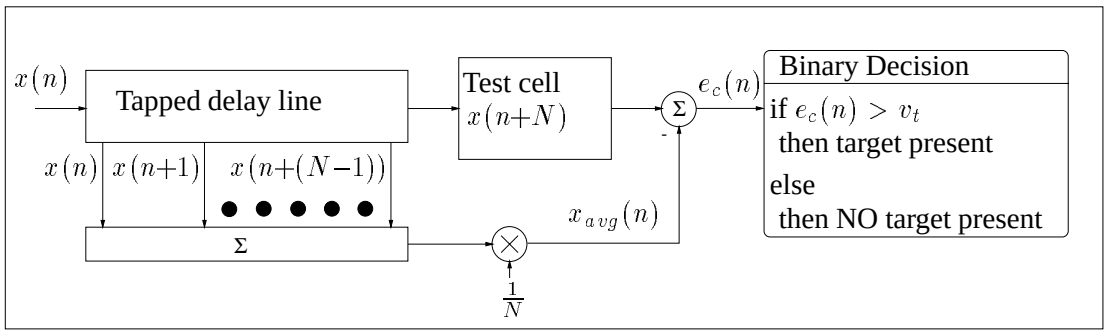


Figure 7.3: *Cell-averaging CFAR. A sliding window of length N is used to estimate the average clutter value, $x_{avg}(n)$. To determine if a target is present in the test cell (*i.e. $x(n+N)$*), the test cell sample is compared with $x_{avg}(n)$. If $x(n+N)$ exceeds $x_{avg}(n)$ by greater than v_t, a target is declared to be present, and if it does not, a target is declared to be not present.*

It can be seen from Figure 7.3, that a CA-CFAR is very similar to an N-tap (1-step ahead) LP, except that instead of obtaining a set of N-tap weights (for some desired cost function, *e.g.* mean square error), the weights of the CA-CFAR are simply $\frac{1}{N}$. The CA-CFAR can be thought of as a crude LP.

More often than not, a CFAR will be designed with a tapped delay line either side of the test cell. This design is very similar to the forward-backward predictors discussed in section 7.4. For the purposes of this thesis, a CA-CFAR with a tapped delay line before the test cell will be referred to as, simply, a CA-CFAR. A CA-CFAR with a tapped delay line either side of the test cell will be referred to as a forward-backward CA-CFAR (FB-CA-CFAR). The structure of a FB-CA-CFAR is given in Figure 7.4.

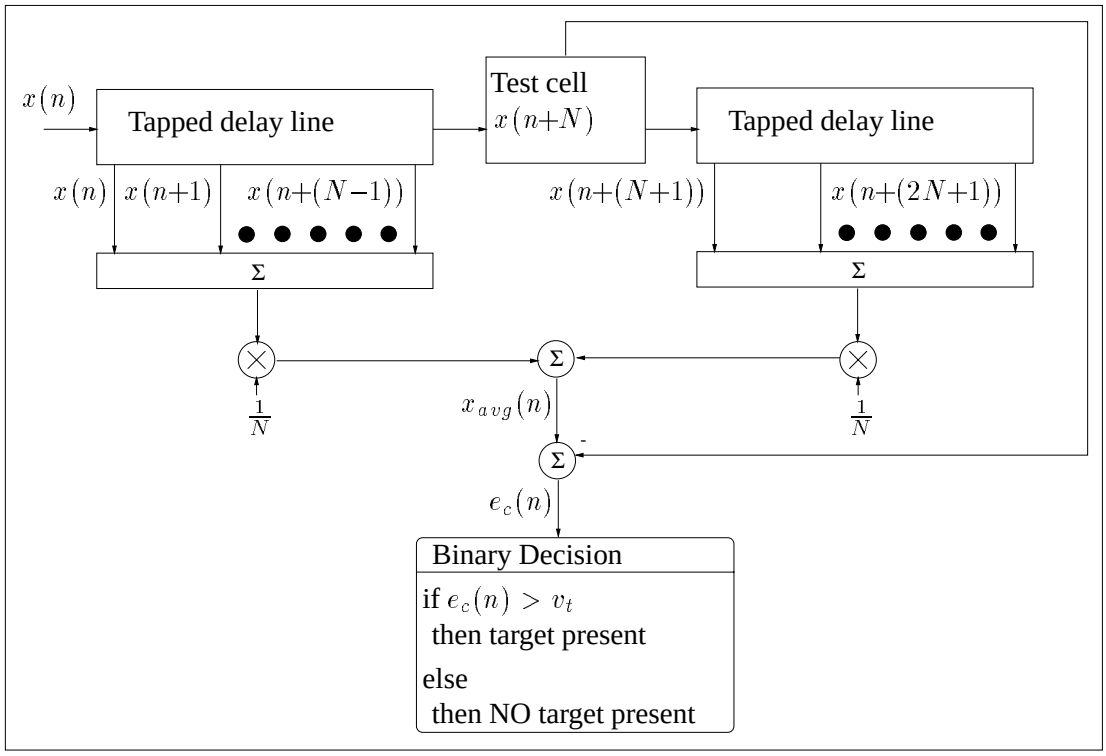


Figure 7.4: *Forward-backward CA-CFAR. A sliding window of length $2N$ (N either side of the test cell) is used to estimate the average clutter value, $x_{avg}(n)$. To determine if a target is present in the test cell (*i.e. $x(n+N)$*), the test cell sample is compared with $x_{avg}(n)$. If $x(n+N)$ exceeds $x_{avg}(n)$ by greater than v_t, a target is declared to be present, and if it does not, a target is declared to be not present.*

Guard cells [6] can be used between the test cell and the tapped delay lines (or CFAR window), in case a target signal is spread across many samples, and is not local to just one sample.

However, for the controlled detection analysis in this thesis, this precaution was not required.

Note that in a similar fashion to the thresholding technique employed for the predictor-detectors, the absolute value of the test cell sample minus the CFAR estimate of the average clutter value, (*i.e.* $|x(n+N) - x_{avg}(n)|$), was used to set the threshold of the CFAR detector. This absolute value will be referred to as $e_C(n)$ hereafter.

To obtain ROC plots for the CA-CFAR and FB-CA-CFAR detectors which could be compared with those for the predictor-detectors, the following approach was adopted.

Detection analysis using a CFAR detector:

- The sea clutter data was divided into the same training (of length Y samples) and non-training (of length X samples) data sets, as for the predictor-detector simulations.
- For any given SCR, the target data generated for use by the predictor-detectors was used by the CFAR detectors.
- The $e_C(n)$ values of the CFAR were collected during the processing of the training data set through the CFAR. These values were used to determine a suitable range of values for the threshold level, over which to carry out the detection analysis on the non-training data set. This was done as follows. The number of different v_t values to use was selected, *i.e.* YY. The maximum and minimum training data set $e_C(n)$ values were determined, and these will be referred to as e_{Cmax} and e_{Cmin}, respectively. The range of v_t values used was, $\{e_{Cmin}, \left[e_{Cmin} + \left(\frac{e_{Cmax} - e_{Cmin}}{YY}\right)\right], \ldots, \left[e_{Cmin} + (YY-1)\left(\frac{e_{Cmax} - e_{Cmin}}{YY}\right)\right]\}$.
- P_{fa} estimates for a given v_t value were obtained by dividing the number of times a $e_C(n)$ value from the non-training data set crossed v_t, by the total number of non-training data set $e_C(n)$ values.
- P_d estimates were obtained for a given v_t value and target sample $t(n)$ by dividing the number of times a non-training $e_C(n)$ value plus target sample (*i.e.* $e_C(n) + t(n)$), crossed v_t, by the total number of non-training $e_C(n)$ values. To obtain an average P_d estimate, for any v_t setting, the P_d estimates for each of the X different target samples were averaged.
- Using the estimated P_d and P_{fa} values, a ROC could be plotted for the CFAR detector, which could be compared with the ROC's for the predictor-detectors.

7.3.4 Fixed threshold detector

Another standard radar target detection technique is the fixed threshold detector. This detector selects a value, usually chosen with knowledge of the clutter statistics involved, to obtain a tolerable number of false alarms. This threshold value is then fixed. This type of detector is optimal in uncorrelated data [39]. However, in correlated clutter CFAR detectors can be used to obtain better detection performance than a fixed threshold detector [66].

To obtain ROC plots for a fixed threshold detector which could be compared with those for the predictor-detectors and CFAR detectors discussed above, the following approach was adopted.

Detection analysis using a fixed threshold detector:

- The sea clutter data was divided into the same training (of length Y samples) and non-training (of length X samples) data sets, as for the predictor-detector and CFAR detector simulations.
- For any given SCR, the target data generated for use by the predictor-detectors was used by the fixed threshold detector.
- The clutter samples $x(n)$ in the training data set were used to determine a suitable range of values for the threshold level, over which to carry out the detection analysis on the non-training data set. This was done as follows. The number of different v_t values to use was selected, *i.e.* YY. The maximum and minimum training data set values were determined, and these will be referred to as x_{max} and x_{min}, respectively. The range of v_t values used was, $\{x_{min}, \left[x_{min} + \left(\frac{x_{max}-x_{min}}{YY}\right)\right], \ldots, \left[x_{min} + (YY-1)\left(\frac{x_{max}-x_{min}}{YY}\right)\right]\}$.
- P_{fa} estimates for a given v_t value were obtained by dividing the number of times a clutter sample from the non-training data set crossed v_t, by the total number of non-training data set clutter samples.
- P_d estimates were obtained for a given v_t value and target sample $t(n)$ by dividing the number of times a non-training clutter sample plus target sample (*i.e.* $x(n) + t(n)$), crossed v_t, by the total number of non-training clutter samples. To obtain an average P_d estimate, for any v_t setting, the P_d estimates for each of the X different target samples were averaged.
- Using the estimated P_d and P_{fa} values, a ROC could be plotted for the fixed threshold

detector, which could be compared with the ROC's for the predictor-detectors, and the CFAR detectors.

7.4 The importance of learning a signal's underlying dynamics for radar target detection

From the prediction results presented in Chapter 5, it was established that a LP could predict the sea clutter data sets described in Appendix A, as well as, or better than, a NLP, in terms of NMSE. Furthermore, there was no evidence found of underlying nonlinear dynamics in the clutter data sets, using recursive prediction. Where recursive prediction did work, for the Dawber VV in-phase channel data set, the recursively predicted data had different properties to those of the original data, see Chapter 5. Despite this lack of evidence of chaotic behaviour, an interesting question was considered during the prediction analysis of the clutter data sets. The question is as follows: is capturing a signal's underlying dynamics (assuming it *has* underlying dynamics) more important than NMSE, in terms of predictor-detector performance? In other words, which would work better, a NLPD that consisted of a NLP that had been trained so as to capture a chaotic signal's underlying dynamics, or one that consisted of a NLP that had been trained to perform better, in terms of NMSE, than the NLP that had learnt the underlying dynamics. The answer to this question may seem intuitively obvious: surely the predictor that could predict the clutter data with the smallest NMSE value would be better able to distinguish between predictor error samples and predictor error plus target samples. However, it was decided to investigate this more thoroughly using detection simulations. This was done using a signal known to be chaotic, and therefore known to have associated underlying dynamics. The signal used was the noisy Lorenz data, described in section 4.7.3. This data consisted of the noise-free Lorenz data described in section 4.7.1, with Gaussian white noise added to it. The SNR was 25dB. The NRBFNP's used in section 4.7.4 for the prediction analysis of this data were used for the detection analysis of the noisy Lorenz data. A NRBFNP-RSC with an embedding dimension of 7, embedding delay of 3, 400 kernels, and a training length of 2000 samples had been successfully used to capture the underlying dynamics of the noisy Lorenz data, as described in section 4.7.4. A NRBFNP-RSC with an embedding dimension of 7, an embedding delay of 1, 400 kernels, and a training length of 2000 samples was not able to successfully capture the underlying dynamics of the noisy Lorenz data, but it did manage to perform slightly better, in terms of NMSE, than the NRBFNP-RSC that had managed to capture the underlying

dynamics, see Table 4.3.

Detection analysis was carried out on the noisy Lorenz data, using a Swerling I target, for SNR's (where the target is the signal, and the noisy Lorenz data is the noise), of 0.2dB and 10dB. The following detectors were used. In addition to the NRBFN predictor-detectors (which used the NRBFNP-RSC's discussed above), a linear predictor-detector with 30 taps, a VSF predictor-detector with an embedding dimension of 10 and embedding delay of 1 sample, a CA-CFAR of size[1] 1, a FB-CA-CFAR of size[2] 1+1, and a fixed threshold detector were all used. The following detection simulation parameters were common to all the detectors.

- Training data set length (Y): 2000 samples.
- Non-training data set length (X): 35,000 samples.
- Number of target samples (ZZ): 35,000.
- Number of different v_t values (YY): 200

The detection results for the above simulations are plotted in Figure 7.5. The ROC's in Figures 7.5(a) and 7.5(c) show the full range of results obtained in the detection analysis of the noisy Lorenz data. However, because only 35,000 noisy Lorenz samples were used, there is an inherent limit to the range of P_{fa} values that can be investigated. Furthermore, with the range investigated in Figures 7.5(a) and 7.5(c), there is a level of confidence associated with each P_{fa} estimate: for example, with the fixed threshold detector, and a non-training data set length of 35,000 samples, a P_{fa} of 2.857×10^{-5} was estimated using only 1 false alarm. Relying on one occurrence of an event is not statistically sound. In the absence of more data[3], a ROC can be edited using a P_{fa} *threshold*: below the threshold, ROC results are thought of as being unreliable, and are discarded, above the threshold the ROC results are thought of as being more reliable than those results below the threshold, and these results are used in the edited ROC. The P_{fa} threshold chosen for the work in this thesis was the P_{fa} associated with at least 100 false alarms: *i.e.* P_{fa} values estimated using less than 100 false alarms were regarded as being

[1] Through trial and error, 1 was found to be the optimal length for the CA-CFAR

[2] *i.e.* 1 element in the forward section, and 1 element in the backward section. As with the CA-CFAR, this size was found to be optimal.

[3] More noisy Lorenz data could be generated, however, data record length is a constraint for the sea clutter data sets discussed in Appendix A, therefore the implications of such short data records are investigated at this juncture.

unreliable. This choice was an arbitrary one. A higher value could be used to increase confidence in the ROC results. Edited ROC results for Figures 7.5(a) and 7.5(c) are given in Figures 7.5(b) and 7.5(d), respectively. For the results in Figure 7.5 a P_{fa} threshold of 0.0029 was used.

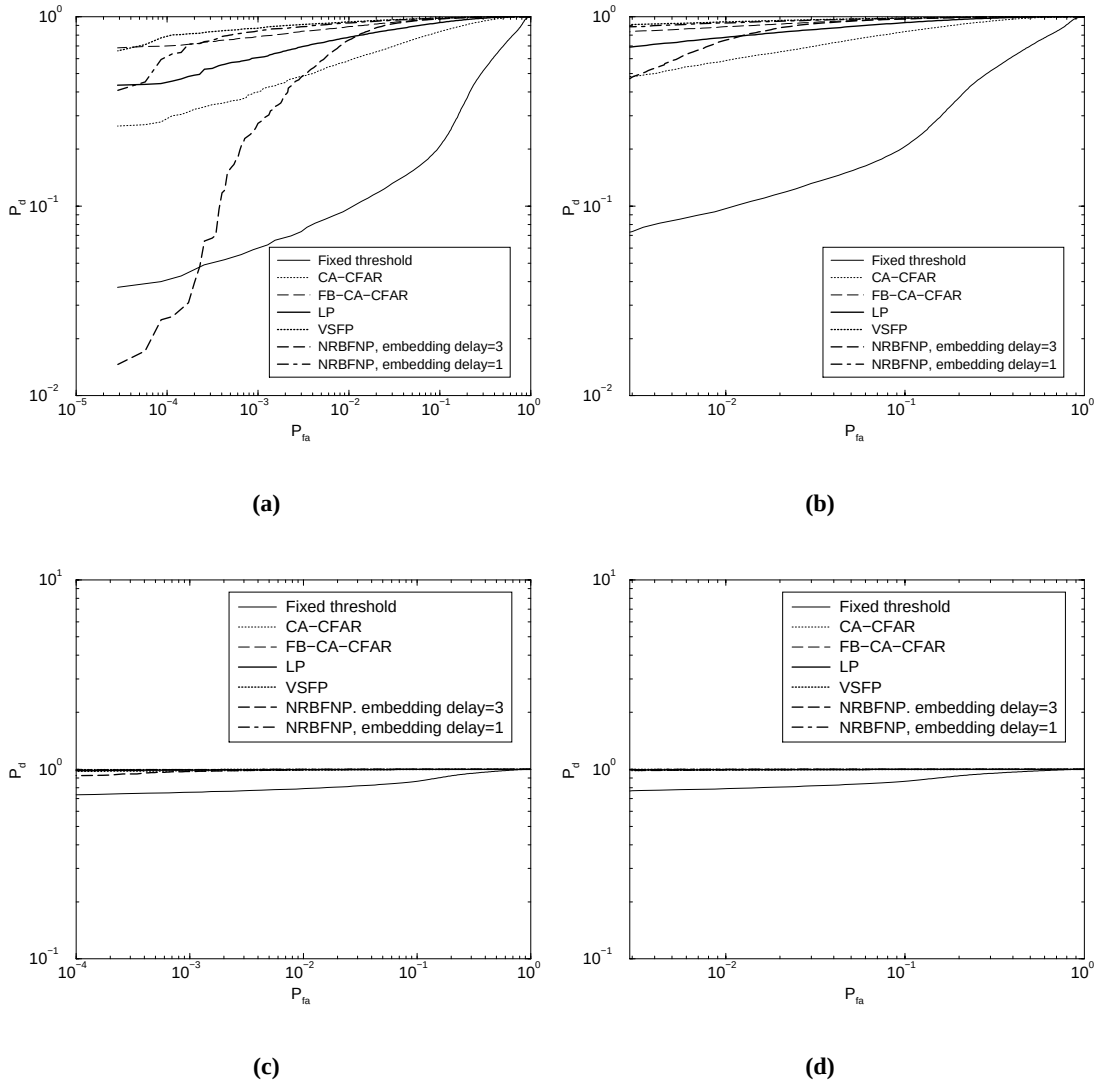


Figure 7.5: *Detection analysis using noisy Lorenz data: (a) ROC for SNR=0.2dB, (b) edited ROC for SNR=0.2dB, (c) ROC for SNR=10dB, and (d) edited ROC for SNR=10dB.*

For the case of a SNR of 0.2dB, it can be seen that the NRBFNP-detector with an embedding delay of 1 sample performed as well as, or better than, the NRBFNP-detector with an embedding delay of 3 samples. In other words the NRBFNP-detector with the better NMSE performed better than the NRBFNP-detector that was able to capture the underlying dynamics of the noisy Lorenz data. From this evidence, given the choice between NMSE and capturing

a signal's underlying dynamics, in terms of detection performance, the predictor-detector with the better NMSE should be selected. To further consider the importance of NMSE, it is necessary to study the prediction performance of the predictor-detectors and CFAR detectors[4] on the training and non-training data sets, as given in Table 7.2.

Predictor	**Training NMSE [dB]**	**Non-training NMSE [dB]**
NRBFNP, embedding delay=1	-23.63	-21.19
NRBFNP, embedding delay=3	-21.59	-18.70
VSFP	-23.38	-21.79
LP	-15.77	-15.52
FB-CA-CFAR	-19.90	-19.92
CA-CFAR	-11.02	-11.08

Table 7.2: *Prediction NMSE performance for predictors on noisy Lorenz data.*

The rule that a better (*i.e.* more negative) NMSE value makes for a better predictor-detector, or CFAR detector, seems to apply in Figure 7.5(b), except for the case of the NRBFNP with an embedding delay of 3 samples (*i.e.* the network that managed to capture the noisy Lorenz data's underlying dynamics). This NRBFNP managed to achieve a better non-training data set NMSE value than the LP, but for low P_{fa} values the NRBFNP-detector with an embedding delay of 3 samples performed more poorly than the LP-detector. The reason for the poorer performance of the NRBFNP-detector, with an embedding delay of 3 samples, can be seen by considering the predictor errors plotted in Figure 7.6.

The reason why the NRBFNP-detector with an embedding delay of 1 sample performed better than the NRBFNP-detector with an embedding delay of 3 samples, and the LP-detector, is because it produced smaller errors than the other predictor-detectors, which allowed better distinction between error plus target samples and error only samples to be made. The LP-detector was able to perform better than the NRBFNP-detector with an embedding delay of 3 samples, due to the same reason. Although the NRBFNP-detector with an embedding delay of 3 samples had a better *overall* NMSE value, it contained many error samples that were larger than those of the LP-detector, which resulted in poorer performance than the LP-detector, at low P_{fa} values.

Before summarising the results for Figures 7.5(a) and 7.5(b), it should be pointed out that for the case of a SNR=10dB, no discernible performance difference was observed between the

[4] As mentioned in section 7.3.3, a CFAR can be considered as a crude predictor, and consequently a NMSE figure can be worked out for it, using the CFAR error values, $e_C(n)$, see section 7.3.3.

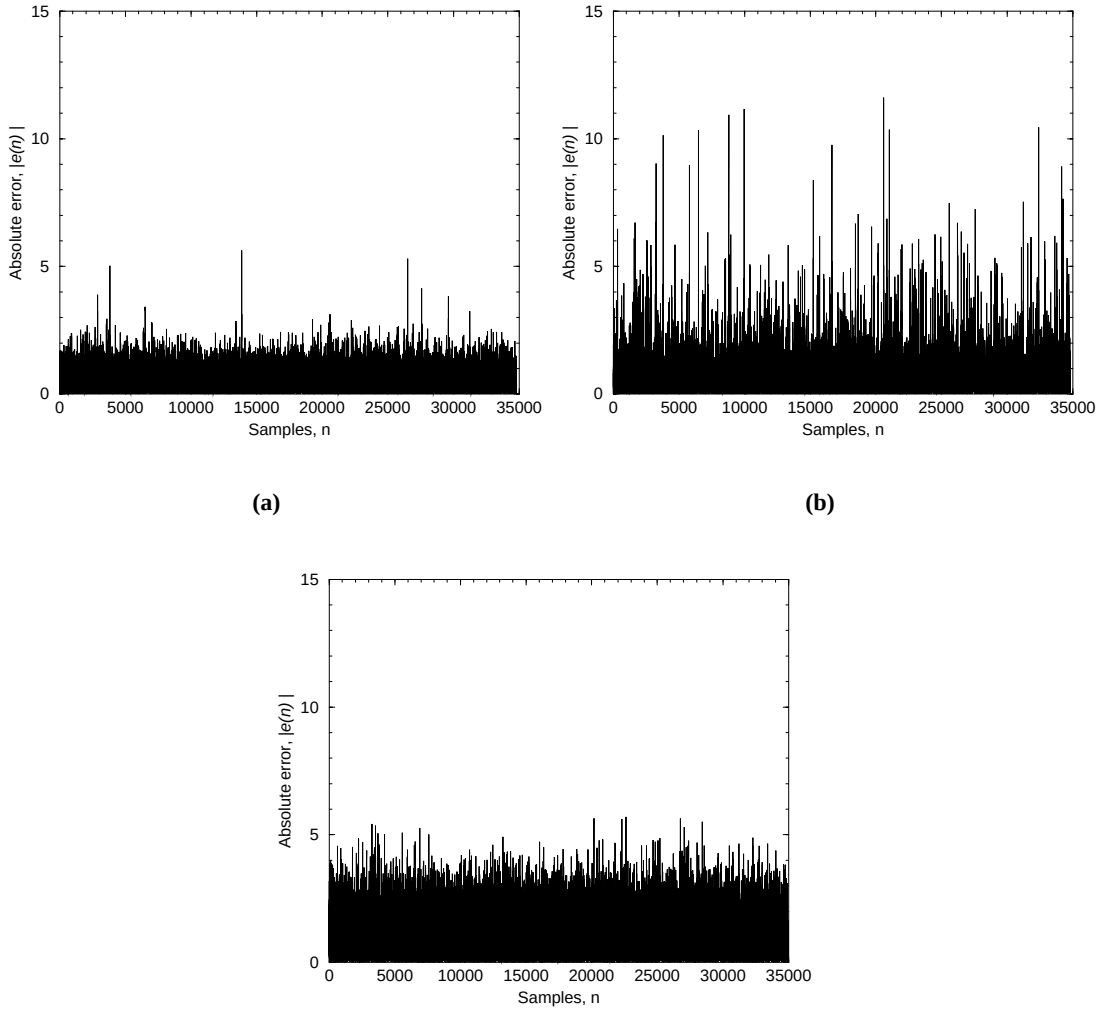


Figure 7.6: *Predictor errors from the target detection analysis against a noisy Lorenz background: predictor errors for (a) a NRBFNP-detector with an embedding dimension of 7 and an embedding delay of 1 sample, (b) a NRBFNP-detector with an embedding dimension of 7 and an embedding delay of 3 samples, and (c) a LP-detector with 30 taps.*

predictor-detectors and CFAR detectors in the edited ROC of Figure 7.5(d).

To summarise the above results, for the case of training a NLP for use in a NLP-detector, it would appear that the smallest NMSE criteria would be preferred to the criteria of training a NLP to capture a signal's underlying dynamics, given that the network that had learnt the underlying dynamics had a poorer NMSE than the network which had not learnt the underlying dynamics of the signal involved. However, as in the case of the NRBFNP-detector with an embedding delay of 3 samples, using only NMSE alone as a guide to the performance of a predictor-detector can be deceiving. It may be that in order to use NMSE as a guide to a predictor-detector's performance, only predictor-detectors using the same embedding delay should be compared. However, as the sea clutter data sets in this thesis were not found to be chaotic, or nonlinearly predictable (see Chapter 5), this matter was not pursued further.

7.5 Detection analysis against a sea clutter background

Detection analysis is now presented using the sea clutter data sets described in Appendix A. In Chapter 5, the prediction analysis showed that each clutter data set had a linear predictor function. Evidence was also presented which showed that a stochastic, compound Gaussian (*i.e.* locally Rayleigh) model was an appropriate model for the clutter data sets analysed, and that speckle (or Rayleigh) correlations were the source of the linear predictability of the clutter data sets. The implications of the prediction results presented in Chapter 5 are now discussed by presenting detection results against a sea clutter background, using a correlated clutter data set and an uncorrelated clutter data set. The correlated data set used was the $12ms^{-1}$ gate 14 wavetank data set. The correlation properties of this data set are given in Figure 5.15(a). The uncorrelated data set used for this detection analysis was the $4ms^{-1}$ gate 15 wavetank data set. The complex ACF for the coherent $4ms^{-1}$ gate 15 wavetank data set is given in Figure 5.17. The ACF of the amplitude signal of the $4ms^{-1}$ gate 15 wavetank data set, is given in Figure 7.7.

Two different SCR's were considered for this analysis, a small SCR of 0.2dB, and a larger one of 10dB. As in section 7.4, a Swerling I target was used. The wavetank clutter data sets each only consisted of 30,000 samples, which has obvious constraints on the range of P_{fa} values that can be considered. As discussed in section 7.4, there were concerns about the reliability of ROC results for small P_{fa} values, and therefore, as in section 7.4, ROC editing was employed,

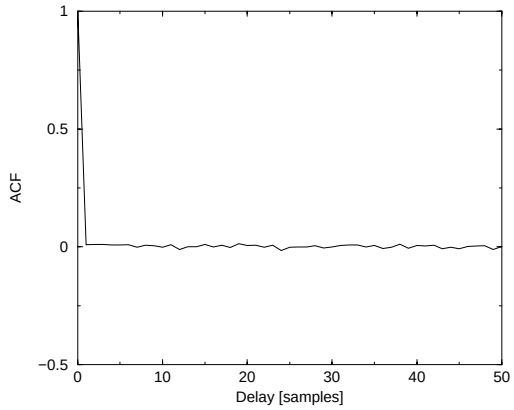


Figure 7.7: *The ACF of the amplitude signal of the $4ms^{-1}$ gate 15 wavetank data set.*

using a P_{fa} threshold. Results will be presented for both un-edited, and edited ROC's, but the discussion of results will be limited to the (more reliable) edited ROC's.

7.5.1 Detection analysis using the amplitude $12ms^{-1}$ gate 14 wavetank data set

Prediction analysis was reported in section 5.2, for the $12ms^{-1}$ gate 14 wavetank data set. Evidence was presented which suggested that this wavetank data set has a linear predictor function. This evidence was gathered by comparing the performance of a NRBFNP with a simple 10 tap LP: a wide range of NRBFNP free parameters were used in this comparison. The free parameters of the predictors used in the predictor-detectors, for the detection analysis of the amplitude $12ms^{-1}$ gate 14 wavetank data set, were as follows. A LP with 20 taps, a NRBFNP with an embedding dimension of 20, an embedding delay of 1 sample, and a VSFP with an embedding dimension of 10, and an embedding delay of 1 sample. The following detection simulation parameters were used for each detector.

- Training data set length (Y): 4000 samples.
- Non-training data set length (X): 26,000 samples.
- Number of target samples (ZZ): 26,000.
- Number of different v_t values (YY): 200

The P_{fa} threshold used to edit the ROC's for this analysis was 0.0038. NMSE values for the predictor-detectors and CFAR detectors are given in Table 7.3.

From the prediction results obtained using the non-training data set, listed in Table 7.3, it can be seen that the LP performed better than the other predictors[5] listed. Using NMSE as a guide to predictor-detector performance, it would be assumed that the LP-detector would perform better than any of the other predictor-detectors or CFAR detectors. The LP-detector did perform, for both small and large SCR values, better than any of the other detectors considered, see the ROC's in Figure 7.8.

As can be seen from Figure 7.8, the LP-detector performed better than the other predictor-detectors, and better than the standard CFAR and fixed threshold detectors. The preferred choice of detection strategy to deal with the correlated $12ms^{-1}$ gate 14 wavetank clutter data set would therefore be a LP-detector.

7.5.2 Detection analysis using the amplitude $4ms^{-1}$ gate 15 wavetank data set

The same predictor parameters, and detection simulation parameters, as used for the detection analysis of the $12ms^{-1}$ gate 14 wavetank data set, were used for the detection analysis of the uncorrelated amplitude $4ms^{-1}$ gate 15 wavetank data set.

The P_{fa} threshold used to edit the ROC's for this analysis was 0.0038. NMSE values for the predictor-detectors and CFAR detectors are given in Table 7.4. ROC results are presented in Figure 7.9.

From Figure 7.9 it can be seen that for both cases of SCR there was no significant difference in performance between the predictor-detectors and the CFAR detectors. Moreover, the predictor-detectors and CFAR detectors performed slightly better than the fixed threshold detector at high P_{fa} values for a SCR of 0.2dB, and at low P_{fa} values for a SCR of 10dB. Based on the above information the CA-CFAR detector would be the preferred choice of detection strategy, as it performed as well as the FB-CA-CFAR and the predictor-detectors, and it is computationally cheaper to implement. It should be noted that the reason why the complex ACF and amplitude ACF plots for the $4ms^{-1}$ gate 15 wavetank data set, in Figures 5.17 and 7.7, respectively, appear to be uncorrelated, is due to the fact that the speckle, or Rayleigh, component of the clutter is uncorrelated, assuming a compound Gaussian model is a suitable model for this data. However, after analysing the data set, and assuming that, in particular, the compound K-distribution is an appropriate model for this data, it was found that the gamma, or underlying, component of

[5]The CFAR detectors are considered as crude linear predictors, as already mentioned in section 7.3.3.

Predictor	Training NMSE [dB]	Non-training NMSE [dB]
NRBFNP	-10.800497	-7.510112
VSFP	-13.506653	-9.142045
LP	-12.686211	-14.919746
FB-CA-CFAR	-7.304652	-8.160810
CA-CFAR	-5.199399	-6.026072

Table 7.3: *Prediction NMSE performance for predictors on $12ms^{-1}$ gate 14 wavetank data set.*

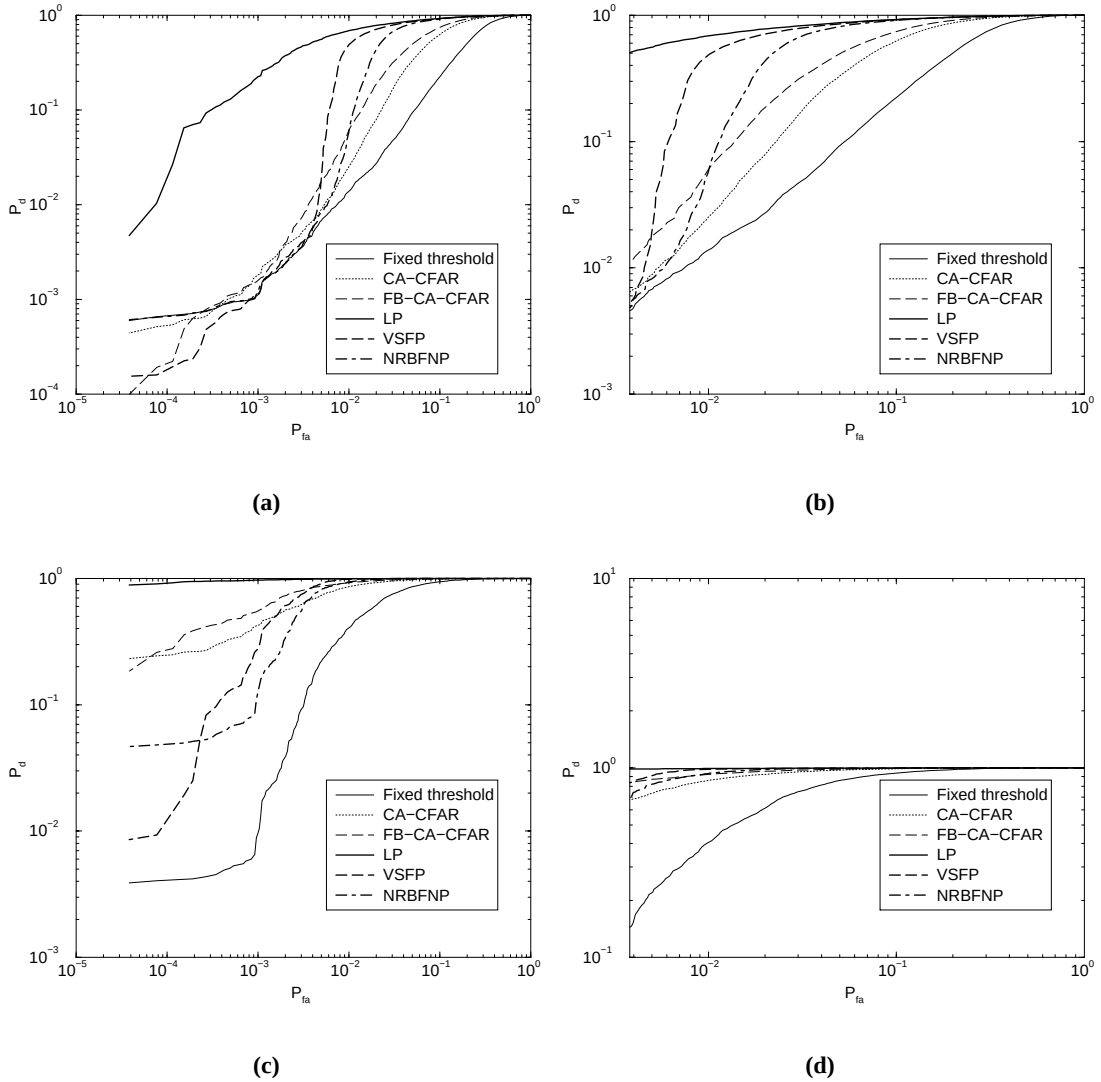


Figure 7.8: *Detection analysis using $12ms^{-1}$ gate 14 data set: (a) ROC for SCR=0.2dB, (b) edited ROC for SCR=0.2dB, (c) ROC for SCR=10dB, and (d) edited ROC for SCR=10dB.*

Predictor	Training NMSE [dB]	Non-training NMSE [dB]
NRBFNP	-0.092089	0.112609
VSFP	-0.345078	0.355768
LP	0.189906	0.212722
FB-CA-CFAR	0.088573	0.084429
CA-CFAR	0.212130	0.195670

Table 7.4: *Prediction NMSE performance for predictors on $4ms^{-1}$ gate 15 wavetank data set.*

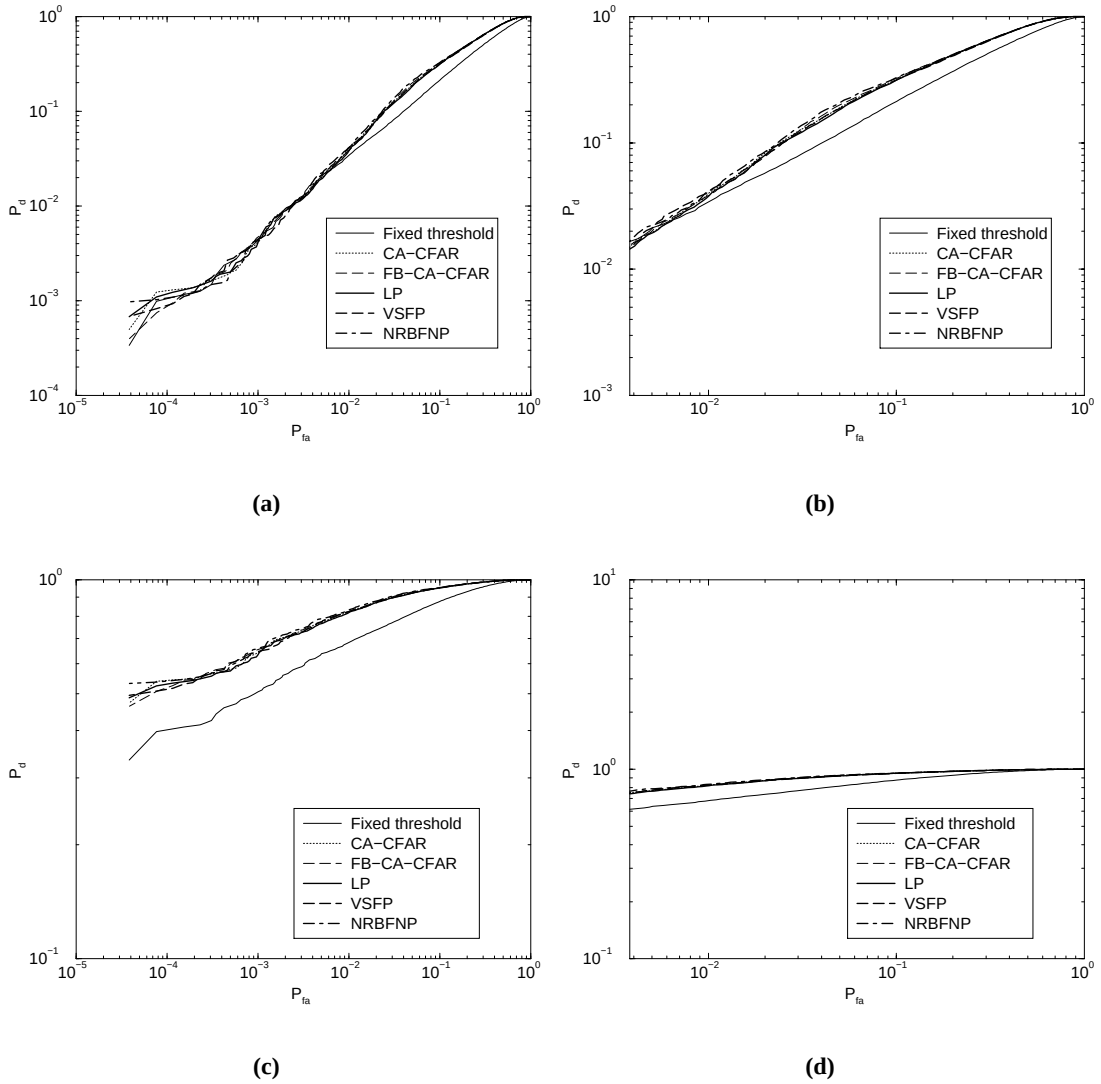


Figure 7.9: *Detection analysis using $4ms^{-1}$ gate 15 data set: (a) ROC for SCR=0.2dB, (b) edited ROC for SCR=0.2dB, (c) ROC for SCR=10dB, and (d) edited ROC for SCR=10dB.*

the clutter was correlated with a correlation length of approximately 170 samples, see Figure 7.10. A fixed threshold detector is only optimal if both components of the compound model are uncorrelated [39], and this is the reason why the fixed threshold detector was not optimal for detection against the $4ms^{-1}$ gate 15 wavetank clutter data set.

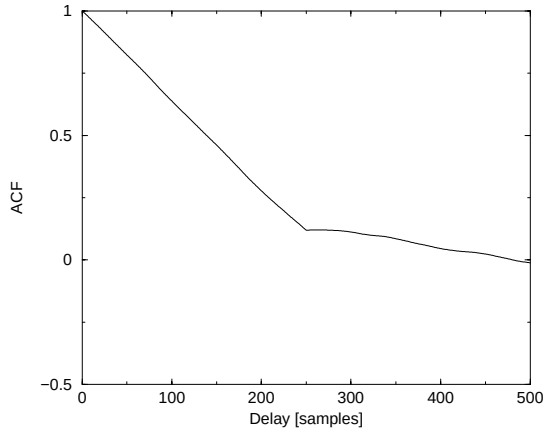


Figure 7.10: *The ACF of the gamma component of the $4ms^{-1}$ gate 15 wavetank data set, obtained using a sliding window of length 250 samples to obtain gamma sample estimates, and then finding the ACF of these gamma samples.*

7.6 Summary

Detection results were presented in this chapter which showed that capturing a chaotic signal's underlying dynamics is not as important as obtaining the best possible NMSE value. However, attention was also drawn to the fact that a predictor's NMSE value should not be taken as the absolute guide to how a predictor-detector will perform.

ROC results were presented for the detection of a Swerling I target against a correlated wavetank clutter data set background and an uncorrelated wavetank clutter data set background. These results showed that for the correlated data, a LPD performed better than a NLPD, which supports the fact that this data was shown to be linearly predictable in Chapter 5. The results for the uncorrelated data showed that a CA-CFAR was the preferred detection strategy: the lack of any speckle correlations resulted in poor LP performance (*i.e.* NMSE values of close to 0dB), and as a consequence the LPD was only able to perform as well as the simple CA-CFAR detector. The CA-CFAR was able to exploit gamma component correlation properties, and it performed better than a fixed threshold detector.

The above evidence suggests that, given the fact that a NMSE prediction analysis shows a clutter data set to be linearly predictable, a LPD would be preferred to a NLPD. It may be possible that a simple CA-CFAR performs as well as, and is preferred to, a LPD, depending on the correlation properties of the clutter data encountered.

Chapter 8
Nonlinear cancellation of chaotic and non-Gaussian interference processes

8.1 Introduction

In Chapters 5 and 6 results were presented which suggested that the clutter data sets described in Appendix A have linear predictor functions. In Chapter 7 a detection analysis was carried out which demonstrated that given data sets which are linearly predictable, a linear predictor-detector would be preferred to, and would perform as well as or better than, a nonlinear predictor-detector. Therefore, despite the encouraging results presented in [12, 15, 17], for the application of nonlinear predictor networks to radar detection of targets against a sea clutter background, it was found that the application of such networks for target detection against the clutter data sets described in Appendix A was not justifiable. Furthermore, despite the evidence in [9–21], to suggest that sea clutter is a chaotic process, evidence was presented in Chapter 5 which suggested that the clutter data sets described in Appendix A are best modelled as a compound Gaussian stochastic process.

Despite the unsuccessful application of a nonlinear predictor network to maritime surveillance radar target detection, another nonlinear signal processing technique was investigated, to determine if it could be used to improve radar target detection against a sea clutter background. The nonlinear approach in question was first proposed by Broomhead *et al.* [25] in 1996 to cancel chaotic noise from signals of interest, and will be referred to as Broomhead's filtering method. Broomhead's filtering method is described in section 8.2, and included in this section is the justification for applying the filtering method to non-Gaussian processes as well as to chaotic processes. This is followed by an investigation into using Broomhead's filtering technique to cancel wideband chaotic noise from a sine wave signal of interest, in section 8.3. In section 8.4 a novel approach to the bandstop linear filtering aspect of Broomhead's filter method is presented, for the cancellation of wideband chaotic noise from a narrowband Gaussian signal. The application of Broomhead's filter method to cancelling clutter from a narrowband Gaussian

target signal, using the novel bandstop linear filter method discussed in section 8.4, is discussed in section 8.5. A chapter summary is presented in section 8.6.

8.2 Broomhead's filtering method

8.2.1 Diagram and explanation

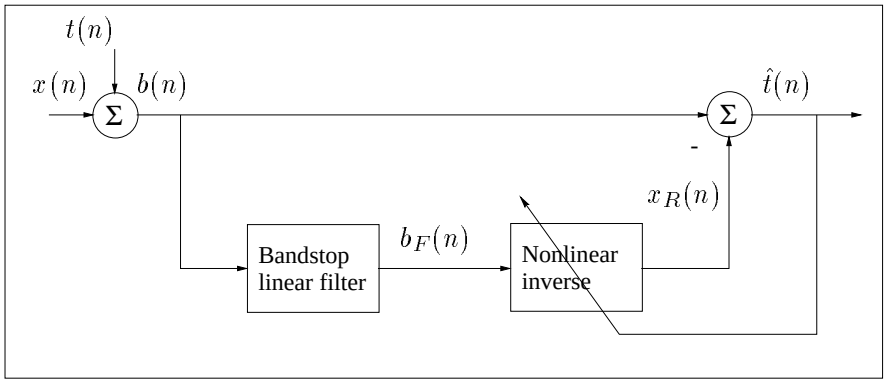
A diagram of the filtering method proposed by Broomhead *et al.* [25] is given in Figure 8.1. A block diagram of Broomhead's filtering method is shown in Figure 8.1(a), with a spectral representation of the filtering operations employed in this filtering technique shown in Figure 8.1(b).

The input signal into the Broomhead filter $\{b(n)\}$ is the linear sum of a narrowband signal of interest $\{t(n)\}$ and a wideband noise process $\{x(n)\}$, *i.e.* $b(n) = t(n) + x(n)$. It is assumed that the spectral properties (*i.e.* the band limits) of the signal of interest are known. In a radar context, the noise process could be sea clutter, and the signal of interest could be reflections from a ship or iceberg. A bandstop linear filter is used to remove the signal of interest from the input signal $\{b(n)\}$. The nonlinear inverse network is used to reconstruct the noise process $\{x(n)\}$ from the output of the bandstop filter $\{b_F(n)\}$. The reconstructed noise process $\{x_R(n)\}$ at the output of the nonlinear inverse can then be subtracted from the input signal $\{b(n)\}$ to obtain an estimate of the signal of interest $\{\hat{t}(n)\}$.

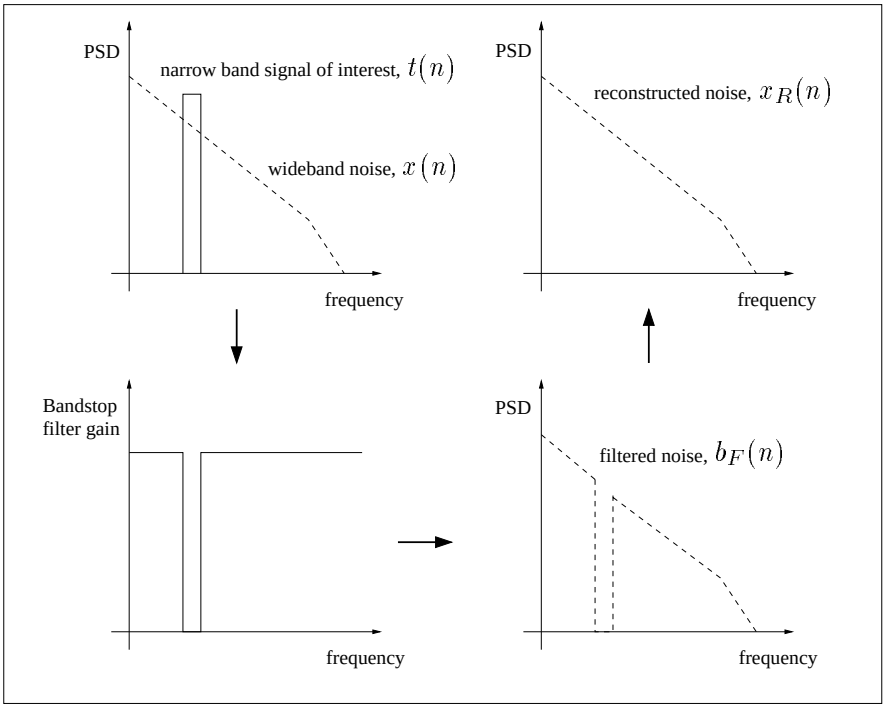
Figure 8.1(b) depicts a case where the bandstop filter is orthogonal to (*i.e.* completely removes) the signal of interest, and where the nonlinear inverse manages to reconstruct, perfectly, the noise process, so that the signal of interest may be obtained, exactly, with no errors.

8.2.2 Linear bandstop filter

The filtering method depicted in Figure 8.1, was originally intended for application to nonlinear deterministic noise cancellation [25]. In such an application, the correct choice of bandstop filter to use is very important. Essentially, the filter must remove the signal of interest (ideally completely), and it must also preserve the dynamics of the chaotic noise, so that the nonlinear inverse can properly reconstruct the chaotic process. A discussion on the selection of an appropriate bandstop filter for this application is given in [4, 25], however the key points of this discussion are now briefly summarised:



(a)



(b)

Figure 8.1: *Broomhead's filter method, (a) block diagram, (b) spectral representation of the noise reconstruction process.*

- Using a short enough linear FIR filter preserves the dynamics of a chaotic signal.
- If the order of a linear FIR filter is too high, the dynamics of a chaotic signal can be changed.
- The higher the order of a bandstop FIR filter, the greater the attenuation is in the stopband.
- An infinite impulse response (IIR) filter changes the dynamics of a chaotic signal, as it has its own associated dynamics.

Clearly, to preserve the dynamics of a chaotic signal, a low order FIR would be preferred. However, there is a tradeoff between signal suppression, and dynamics distortion: a short enough filter may not change the dynamics of a chaotic process, but it may also not adequately suppress the signal of interest. There is ambiguity associated with what length of filter constitutes one that is short enough, *i.e.* of low enough order, to not change the dynamics of a chaotic process. There appears to be no clear cut answer to this question, other than to try a simple trial and error approach, to find a suitable filter length which not only adequately suppresses the signal of interest, but also does not change the dynamics of the noise process.

It should be pointed out that filtering of any kind will distort the dynamics of a chaotic process, however, the aim is to limit this distortion as much as possible so that the nonlinear inverse can produce a reasonable reconstruction of the original chaotic noise process $\{x(n)\}$ from the bandstop filtered chaotic process $\{b_F(n)\}$.

Of course, for the application of interest for this thesis (*i.e.* maritime surveillance radar), the *noise*[1] process is sea clutter. The evidence presented in Chapter 5 of this thesis suggested that the clutter data sets described in Appendix A are not chaotic, but rather, they are each better modelled by a compound Gaussian process. However, as will be discussed in section 8.2.3, the application of Broomhead's filter method to a broader class of non-Gaussian noise processes, rather than just limit its application to situations when the noise process is chaotic, is perfectly justifiable. In applying Broomhead's filter technique to cases when the noise process is not chaotic, but is instead a non-Gaussian stochastic process, it might be reasonable to assume that a discussion of the preservation of a signal's dynamics is irrelevant to the choice of a suitable bandstop filter. However, the distortion of a chaotic signal's dynamics resulting from (linear) filtering can be seen as a more general change in the nonlinear properties of the chaotic signal.

[1] Not to be confused with thermal white noise.

Furthermore, it is suggested that this idea of distorting the nonlinear properties of a signal by filtering can be extended to any non-chaotic noise process, which has nonlinear properties that would allow Broomhead's filter method to perform better than a conventional linear approach. In other words, it may be necessary to exercise the same caution in the selection of a suitable bandstop filter for the application of Broomhead's filtering technique when the noise process is described as a non-Gaussian stochastic process, as is required when the noise process is chaotic.

8.2.3 Why a nonlinear inverse as opposed to a linear inverse?

Linear filtering techniques are unable to relate noise components outside the (target signal) band of interest, with those inside the band, if the noise process is not available both during the training of the linear inverse, and also after training. In a situation where the noise process is only available during training[2], the best a linear inverse noise suppression approach can achieve, is to remove all of the out-of-band noise, still leaving behind the in-band noise, and therefore performing sub-optimally. The interest in using a nonlinear approach, is to try and identify a suitable nonlinear relationship that would allow both the in-band and out-of-band noise components to be suppressed, allowing a nonlinear approach to perform better than a linear one.

Broomhead's filtering method has already been applied to the cancellation of chaotic noise [4, 25]. Results from this previous work are considered further in the following section, section 8.2.4. The application of interest for this thesis is maritime surveillance radar, where the noise process (*i.e.* sea clutter) can be described as a non-Gaussian stochastic process. As already mentioned in section 8.2.2, it is justifiable to apply Broomhead's filtering technique to the broad class of non-Gaussian signals. The reason for this is now given. If a process may be described as a Gaussian stochastic process (correlated or uncorrelated), then all its frequency components are independent, and no part of its spectrum is related to another part [133], and it is therefore impossible to relate out-of-band noise to in-band noise, when the noise process is not known *a priori*. However, for non-Gaussian stochastic signals it may be possible that a nonlinear approach could relate out-of-band noise to in-band noise, and could therefore be used to eliminate in-band noise, and achieve better noise suppression than a linear approach.

[2] This is the case in maritime surveillance radar, where it is assumed sea clutter data can be collected without any target signal present: for instance, the absence of a target signal could be ensured by visually inspecting an area close to the radar.

8.2.4 Previous work

As already mentioned, Broomhead *et al.* [25] applied Broomhead's filter technique to cancel chaotic noise. In an experiment they carried out, involving a sine wave corrupted by chaotic Ikeda [32, 134] noise, Broomhead *et al.* showed that a RBF nonlinear inverse was able to obtain a reasonable performance when the noise process was not known beforehand (*i.e.* the RBF inverse was trained using the noise corrupted signal of interest). They also described the RBF inverse as "indispensable", when the noise process was known beforehand (*i.e.* the RBF inverse was trained using the noise process alone). No linear comparisons were, however, carried out in this work.

Strauch [4] carried out the same experiments as those carried out by Broomhead *et al.* [25], but with linear comparisons. Strauch also proposed three modifications to Broomhead's filtering technique. In the experiment with a sine wave corrupted by Ikeda noise, Strauch found that when the noise process was not known beforehand, there was little or no improvement obtained by using a nonlinear inverse, with respect to using a linear inverse. Also, Strauch reported that very little improvement was observed when the noise process was known beforehand. This result is surprising, and it contradicts the "indispensable" verdict of a nonlinear inverse in such a situation, made by Broomhead *et al.* [25].

8.2.5 Work discussed in this chapter

Due to the surprising result obtained by Strauch [4], for the case when the noise process was known beforehand, it was decided to re-investigate one of Broomhead *et al.*'s [25] chaotic noise cancellation experiments, using linear filtering comparisons. The experiment chosen was for the case where the chaotic noise process was from the Ikeda map, and the signal of interest was a simple sinusoid. The results of this experiment are given in section 8.3. The re-investigation of this experiment lead onto a modified Broomhead filtering approach, which will be discussed in section 8.4. The Broomhead filtering technique of section 8.4 was applied to the cancellation of clutter, and results of this analysis are presented in section 8.5.

8.3 Using a sinusoid corrupted by broadband Ikeda noise to investigate Broomhead's filter method

This experiment was carried out by Broomhead *et al.* [25], and by Strauch [4]. The reasons for re-investigating this experiment were as follows:

- To thoroughly investigate if the disappointing results obtained by Strauch were correct.
- To consider what the effect of using different bandstop linear filters had on the dynamics, or more generally, on the nonlinear properties, of a signal with known dynamics: the Ikeda map.

8.3.1 Signal description

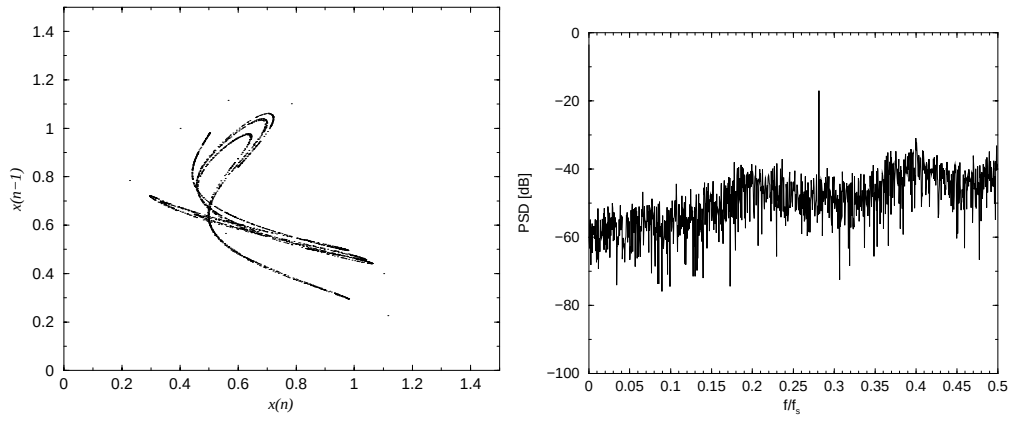
The signal of interest used for this experiment was a sine wave with an amplitude of 0.2, and a normalised frequency f/f_s of 0.28125. The chaotic broadband noise, that was used to corrupt the sine wave, was generated from the complex Ikeda map. This signal will be referred to, simply, as Ikeda noise. The complex Ikeda map is given in equation (8.1).

$$i(n+1) = 1 + \mu_i i(n) \exp\left[\alpha_i - \frac{\beta_i}{1+|i(n)|^2}\right] \tag{8.1}$$

The real part of the complex Ikeda map, $x(n) = \Re\{i(n)\}$, was used to generate the Ikeda noise. In order to generate chaotic behaviour, the complex Ikeda map parameters were chosen to be $\mu_i = 0.7$, $\alpha_i = 0.4$, and $\beta_i = 6.0$. The SNR using the sine wave signal of interest and the Ikeda noise time series, described above, was -2.7dB.

The Ikeda noise attractor is plotted in Figure 8.2(a). The maximum likelihood correlation dimension for the Ikeda noise data was estimated, using 50,000 samples, to be 1.51. Using Takens' embedding delay theorem, this would result in a suitable choice for embedding dimension D_E of the Ikeda noise as $D_E \geq 4.02$. Abarbanel [32] used the global false nearest neighbours method to demonstrate that an embedding dimension of 4, *i.e.* $D_E = 4$, was able to unfold the Ikeda attractor.

A PSD plot of the Ikeda noise corrupted sine wave is given in Figure 8.2(b).



(a) **(b)**

Figure 8.2: *Plots of (a) the Ikeda noise attractor, and (b) the PSD of sine wave corrupted by Ikeda noise.*

8.3.2 Bandpass filtering: performance benchmark

Before applying Broomhead's nonlinear inverse filter method, as well as a linear inverse comparison, to the task of cancelling the Ikeda noise from the sine wave plus Ikeda noise signal, it was decided to use a standard bandpass filter, to establish a performance benchmark by which the nonlinear and linear inverse filter methods could be judged. Infinite impulse response (IIR) Butterworth bandpass filters [127] were used for this task.

The Ikeda noise corrupted sine wave, with an input SNR of -2.7dB, was applied to 6^{th}, 12^{th}, and 18^{th} order Butterworth bandpass filters. Each filter had a passband with a startband frequency of $f/f_s = 0.28$, and a stopband frequency of $f/f_s = 0.2825$ The achieved output SNR for each filter is given in Table 8.1.

Filter order	Input SNR [dB]	Output SNR [dB]
6^{th}	-2.7	23.3
12^{th}	-2.7	24.2
18^{th}	-2.7	24.5

Table 8.1: *Ikeda noise suppression using IIR Butterworth filters.*

10,000 samples were filtered to obtain the results in Table 8.1. Transients of length 600 samples, 1000 samples, and 1200 samples were removed for the 6^{th}, 12^{th}, and 18^{th} order filters, respectively. Figure 8.3, shows the PSD plot of the output of the 18^{th} order bandpass filter.

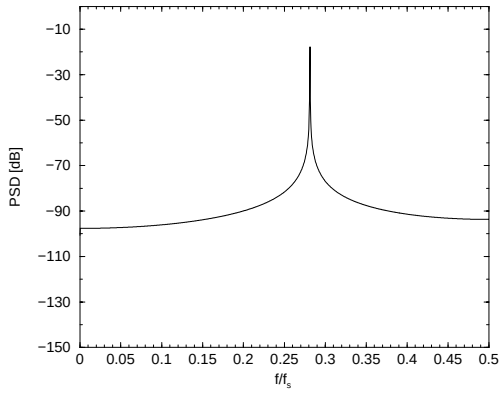


Figure 8.3: *PSD of filtered sine wave corrupted by Ikeda noise data, using an* 18^{th} *order Butterworth IIR filter.*

8.3.3 Broomhead's filtering method using a notch filter

Broomhead's filter method, Figure 8.1, was applied to the (Ikeda) noise corrupted sine wave signal, described in section 8.3.1. A notch [135] filter was used as the bandstop linear filter in Broomhead's filter method. A notch filter is a filter which is used to severely, or *ideally*, completely, suppress a particular frequency, or a number of particular frequencies. This is done by the correct placement of complex conjugate zeros on the unit circle [135]. For example, to suppress a frequency, say f/f_s, a pair of complex conjugate zeros are placed on the unit circle at an angle $2\pi f/f_s$, *i.e.* zeros $= \exp[\pm j2\pi f/f_s]$. The transfer function for a FIR notch filter used to suppress the frequency f/f_s, is given in equation (8.2),

$$H(z) = G(1 - 2\cos(2\pi f/f_s)z^{-1} + z^{-2}) \tag{8.2}$$

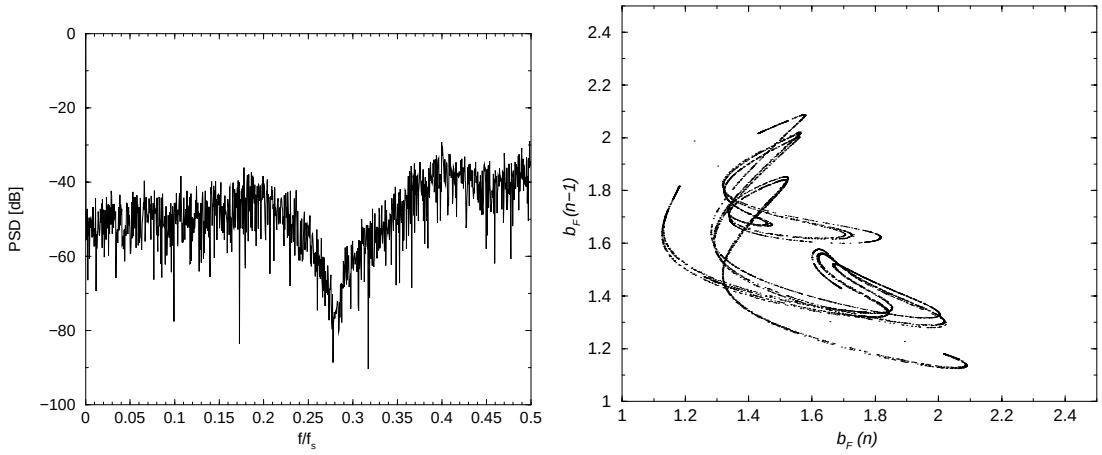
where G is the amplifying term, and for the work described in this section, was set to 1.0.

The work in this section describes the application of Broomhead's filter method to the Ikeda noise corrupted sine wave, for two different scenarios. The first scenario discussed is when only the Ikeda noise corrupted sine wave was available to train the nonlinear inverse. The second scenario discussed is when the Ikeda noise signal was available, separately, from the sine wave signal of interest, and the nonlinear inverse was trained using the Ikeda noise data alone. For both scenarios, a linear inverse comparison was used.

Training the inverse with the signal of interest present

A NRBFN with an embedding dimension of 4 and a training length of 2500 samples was used as the nonlinear inverse. To test the generalisation properties of the nonlinear inverse, testing and validation data sets of length 2500 samples were used. The number of Gaussian kernels used in the NRBFN was varied from 10 to 90 in steps of 10, and from 100 to 800, in steps of 100. A linear comparison was used with a training length of 2500 samples. As with the nonlinear inverse, testing and validation data sets of length 2500 samples were used to judge the generalisation properties of the linear inverse. The number of taps in the linear inverse was varied from 100 to 800, in steps of 100.

The PSD and the attractor plot of the notch filtered noise corrupted sine wave signal are given in Figure 8.4. As can be seen in Figure 8.4(a), the sine wave has been completely removed by

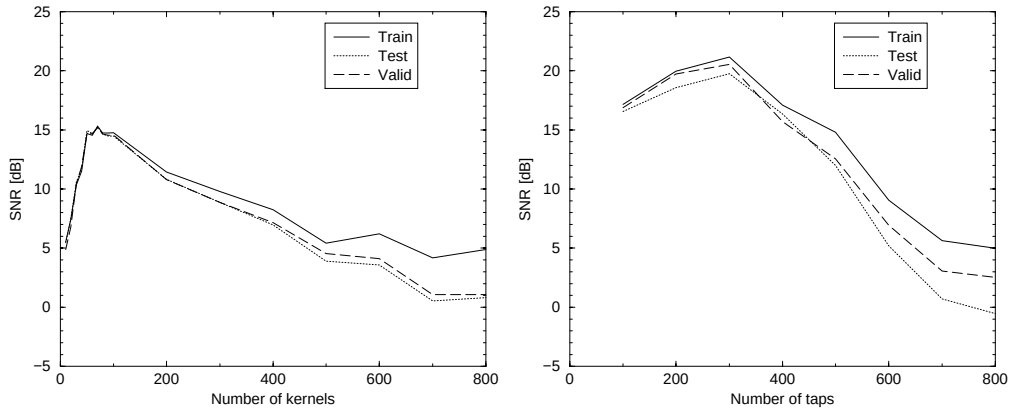


(a) **(b)**

Figure 8.4: *Notch filtered Ikeda noise corrupted sine wave: (a) PSD estimated using 2048 samples, (b) attractor plot.*

the notch filter.

The achieved output SNR's for Broomhead's nonlinear inverse filter method, and for a linear inverse filter are given in Figure 8.5. As can be seen from Figure 8.5, a linear inverse was able to achieve a better output SNR than a nonlinear inverse was able to. However, both inverses performed more poorly than the Butterworth filters mentioned in section 8.3.2. The evidence in Figure 8.5 suggests that when the noise process is not known beforehand, and the inverse must be trained on the noise corrupted signal of interest data, the problem is completely, or mostly,



(a) **(b)**

Figure 8.5: *Output SNR's for (a) nonlinear inverse, (b) linear inverse.*

linear and a linear inverse would be preferred to a nonlinear inverse. Also note from Figure 8.5, that with increasing complexity the linear and nonlinear inverses achieve poorer output SNR's. This is due to the fact that in the training phase, with ever more complexity, the inverse over-fits to the (noise plus signal of interest) training data, which results in the noise corrupted signal of interest being completely cancelled, instead of only cancelling the noise. Due to the over-fitting in the training phase, the generalisation properties of the inverse were poor, and resulted in poor performance in the testing and validation phases. These claims are supported by considering the NMSE results for the inverse networks, shown in Figure 8.6. The NMSE was evaluated as follows,

$$\text{NMSE} = 10\log_{10}\left(\frac{1}{\sigma_b^2 Y}\sum_{n=1}^{Y}(b(n) - x_R(n))^2\right) \tag{8.3}$$

where σ_b^2 is the variance of $b(n)$, the input to Broomhead's filter, $x_R(n)$ is the reconstructed (Ikeda) noise process, at the output of the inverse, and Y is the number of samples over which the NMSE is evaluated. For the work in this chapter, Y samples were used in each phase: training, testing, and validation.

The PSD of the recovered signal of interest, using a nonlinear inverse with an embedding dimension of 4, 100 kernels and a training length of 2500 samples, is shown in Figure 8.7(a). The

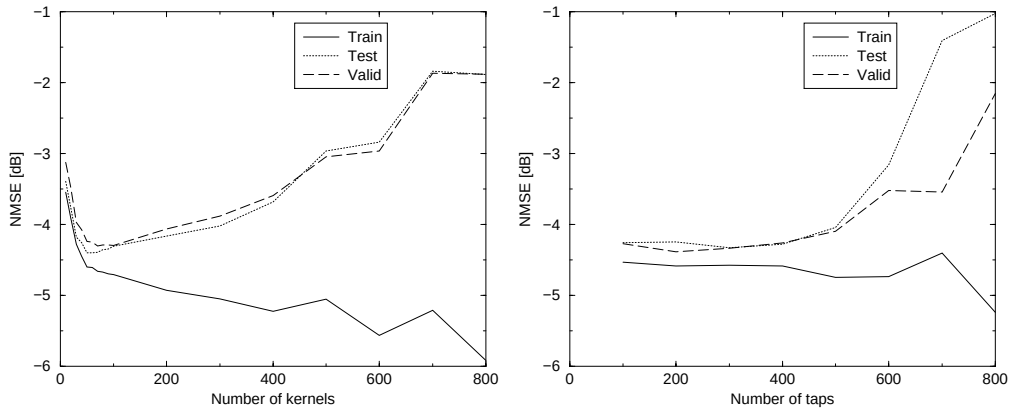


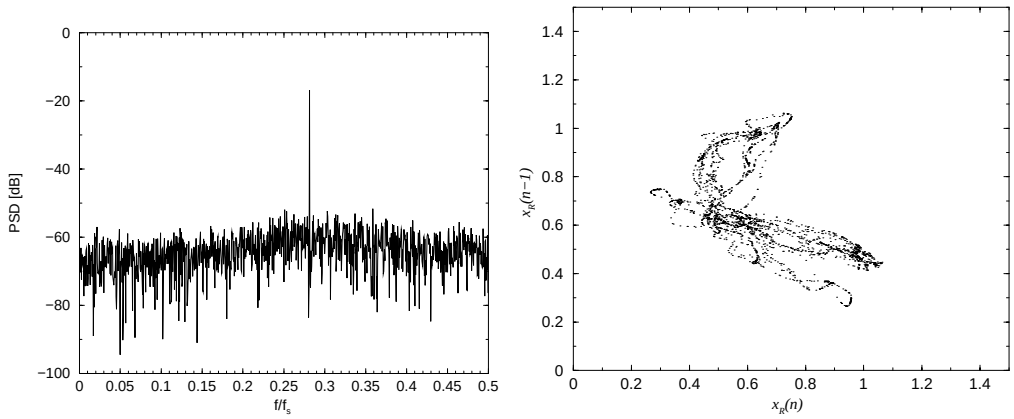
Figure 8.6: *NMSE's for (a) nonlinear inverse, (b) linear inverse.*

attractor of the reconstructed Ikeda noise, obtained using the same nonlinear inverse, is shown in Figure 8.7(b). The results in Figure 8.7 were obtained using the testing data set.

Comparing Figures 8.7(b) and 8.2(a) it can be seen that the reconstructed attractor resembles the original attractor, but is still quite distorted: the presence of the sine wave during the training of the nonlinear inverse hampered the reconstruction of the Ikeda noise.

An embedding dimension of 4 was selected for the nonlinear inverse, as this was the value suggested in [32] for the embedding dimension required to unfold the Ikeda (noise) attractor. Additionally, the minimum value of embedding dimension, estimated using the maximum likelihood correlation dimension (see section 8.3.1) was 4.02, which is very close to the value suggested in [32]. However, a trial and error approach was also conducted, and Broomhead's filter method was re-investigated using embedding dimensions of 1,2,3,5,6,7,8,9, and 10 for the nonlinear inverse. As for the case with an embedding dimension of 4, a NRBFN with a training length of 2500 samples was used, and the number of kernels was varied from 100 to 800 in steps of 100. In general, the best non-training output SNR's were achieved using 100 kernels, and these SNR's are tabulated for each case of embedding dimension in Table 8.2. For each case of embedding dimension investigated, the nonlinear inverse performed more poorly than a linear inverse with 100 taps, see Figure 8.5(b).

To investigate if perhaps the nonlinear inverse could perform better using fewer kernels, and a shorter training length, Broomhead's filter method was re-investigated using a NRBFN with an



(a) **(b)**

Figure 8.7: *Testing data set results for Broomhead's filter method, using a NRBFN inverse with an embedding dimension of 4, 100 kernels, and a training length of 2500 samples: (a) PSD of the recovered signal of interest* $\{\hat{t}(n)\}$ *and (b) the attractor of the reconstructed Ikeda noise data.*

embedding dimension of 4, and 80 kernels. The training length was varied from 500 samples to 2500 samples, in steps of 500 samples. The results from this experiment did not provide any evidence to suggest that the NRBFN could perform better with fewer kernels or shorter training lengths, and moreover, the output SNR's achieved using training lengths of 500, 1000, 1500 and 2000 samples, from this experiment, were poorer than those achieved using a NRBFN inverse, with an embedding dimension of 4, 100 kernels, and a training length of 2500 samples. The output SNR achieved using a NRBFN inverse with 80 kernels and a training length of 2500 samples was better by 0.2dB than that achieved by the NRBFN inverse with an embedding dimension of 4, 100 kernels, and a training length of 2500 samples.

Training the inverse without the signal of interest present

Now results are presented for the scenario where there was access to the Ikeda noise process alone, during the training of the inverse. In other words the inverse was trained using noise only data, without the signal of interest present.

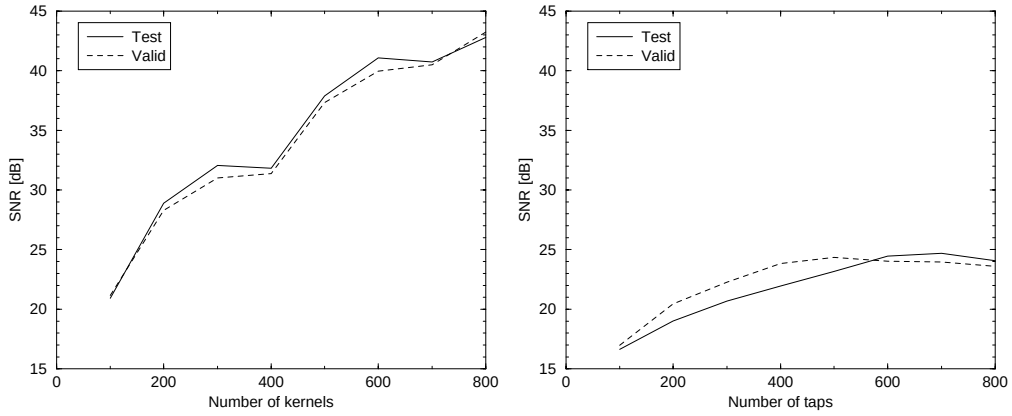
The initial nonlinear inverse experiment that was carried out for the case where the signal of interest was present during the training of the inverse, was repeated without the signal present during training: a NRBFN with an embedding dimension of 4, and a training length of 2500 samples was used. The linear comparison carried out for the case where the signal of interest

N	Training O/P SNR [dB]	Testing O/P SNR [dB]	Validation O/P SNR [dB]
1	-0.3533379510	-0.4681553054	-0.404913140
2	7.4665958247	7.4058175531	7.0103707920
3	14.5318054824	14.4620872437	14.4178614522
4	14.7609502775	14.4826478768	14.5563336367
5	14.7450003202	14.6280824801	14.5742498827
6	14.9939765265	14.7885139129	14.7425518773
7	13.6287974063	13.7103427855	13.8108309824
8	13.5867909473	13.3334075391	13.3883669692
9	11.5073065062	11.2054245621	11.3985337263
10	10.3900814917	9.9181099523	9.9977550096

Table 8.2: *Output SNR's for Broomhead's filter method, using NRBFN nonlinear inverses with 100 kernels, and a training length of 2500 samples, for various embedding dimension, N, values.*

was present during the training of the linear inverse, was repeated without the signal present during training.

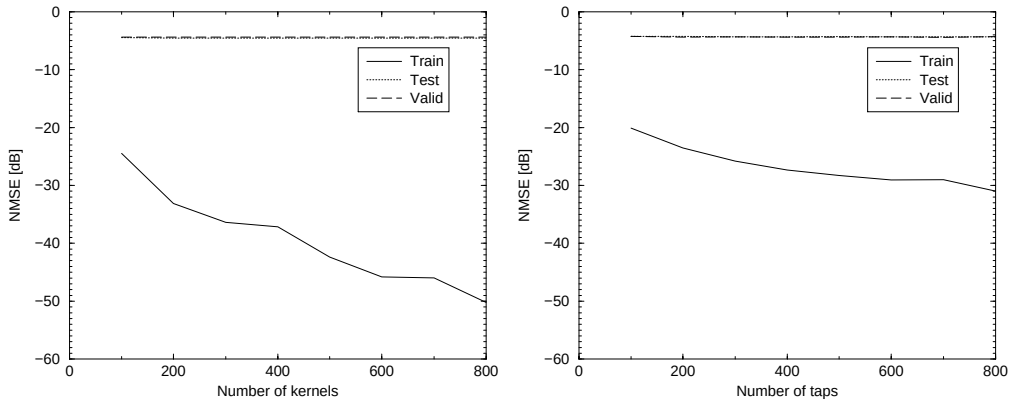
The achieved output SNR's for Broomhead's nonlinear inverse filter method, and for the linear inverse filter comparison are given in Figure 8.8.



(a) **(b)**

Figure 8.8: *Output SNR's for (a) nonlinear inverse, (b) linear inverse.*

As can be seen by comparing Figures 8.8 and 8.5, both the nonlinear and linear inverse techniques performed much better when the signal of interest was not present during the training of the inverse than when it was. With a high enough number of taps, the linear inverse managed to achieve an output SNR (approximately 24dB) comparable with that obtained by the



(a) **(b)**

Figure 8.9: *NMSE's for (a) nonlinear inverse, (b) linear inverse.*

IIR Butterworth filters, which were discussed in section 8.3.2. The nonlinear inverse managed to achieve even better output SNR's than this, reaching a value of approximately 43dB with 800 kernels. The nonlinear inverse was able to perform better than the bandpass filters because it reduced noise that was not suppressed in the IIR Butterworth filter passband, which was $f/f_s = 0.28 \rightarrow f/f_s = 0.2825$.

Figure 8.9 shows the NMSE results for the nonlinear and linear inverses. It can be seen that the nonlinear and linear inverses performed very similarly on the testing and validation data sets. However, the performance of the nonlinear inverse on the training data was substantially better than that of the linear inverse. This training phase performance difference corresponds to a better reconstruction of the Ikeda noise process, and to better testing data set and validation data set output SNR's.

The PSD of the recovered signal of interest, using a nonlinear inverse with an embedding dimension of 4, 800 kernels, and a training length of 2500 samples is shown in Figure 8.10(a). The attractor of the reconstructed Ikeda noise obtained using the same nonlinear inverse is shown in Figure 8.10(b). The results in Figure 8.10 were obtained using the testing data set.

Comparing Figures 8.10(b) and 8.2(a) it can be seen that the reconstructed attractor is indistinguishable from the original attractor. Despite the additional twists and turns, or distortions, of the notch filtered attractor shown in Figure 8.4(b), with respect to the original attractor shown in Figure 8.2(a), the nonlinear inverse was able to accurately reconstruct the Ikeda noise process. In other words, the notch filter was able to sufficiently suppress the signal of interest, without

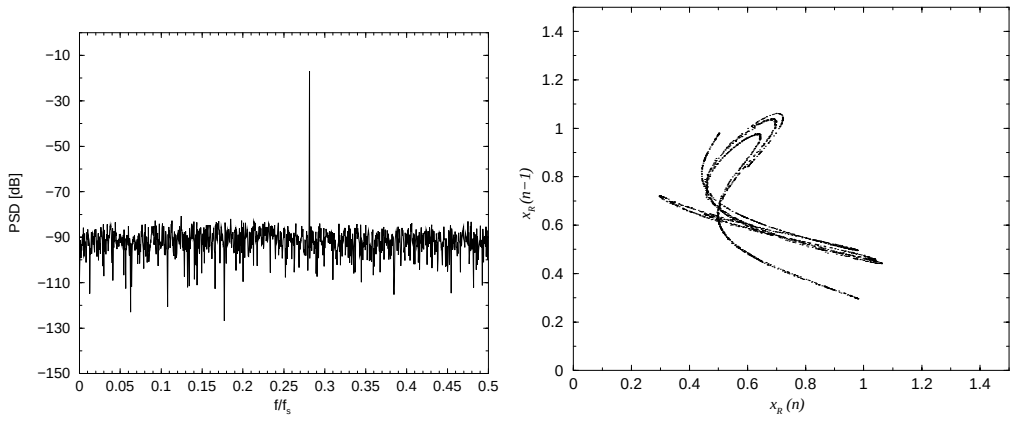


Figure 8.10: *Testing data set results for Broomhead's filter method, using a NRBFN inverse with an embedding dimension of 4, 800 kernels, and a training length of 2500 samples: (a) PSD of the recovered signal of interest* $\{\hat{t}(n)\}$ *and (b) the attractor of the reconstructed Ikeda noise data.*

changing the dynamics, or nonlinear properties, of the Ikeda noise process too much, to allow the nonlinear inverse to accurately reconstruct the Ikeda noise process.

The encouraging performance of the nonlinear inverse which was trained without the signal of interest present, is in marked contrast to the results reported by Strauch [4], for this experiment. Strauch found that there was little improvement (of the order of 1dB) to be gained by training the nonlinear inverse without the signal of interest present, with respect to when it was trained with the signal of interest present. Furthermore, Strauch reported even poorer performance of the nonlinear inverse when trained with the signal of interest present, than has been reported herein. Strauch reported output SNR's of around 7.5dB for the nonlinear inverses he considered, when the signal of interest was present during training. The encouraging performance of the nonlinear inverse, reported in this thesis, which was trained without the signal of interest present agrees with the comment made by Broomhead *et al.* [25], that when a chaotic noise process is known *a priori* a RBFN inverse is "indispensable".

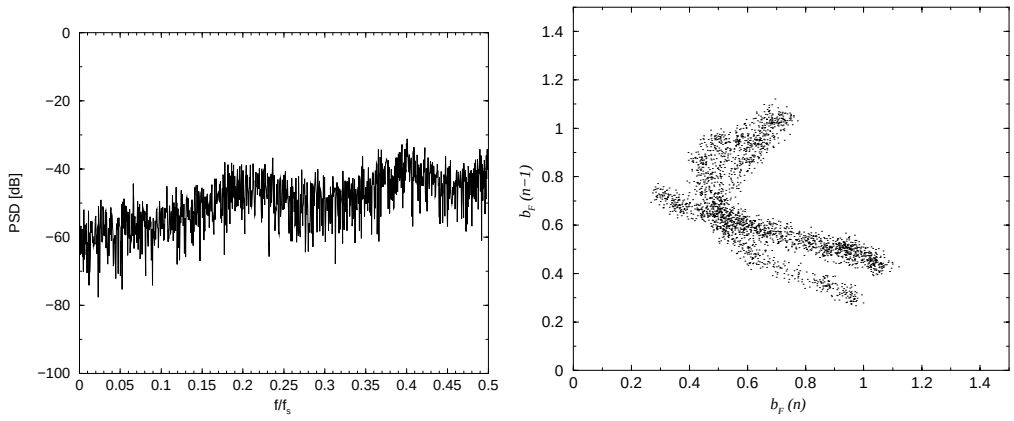
It should be pointed out that, for this simple experiment (using a sine wave as the signal of interest), given enough complexity and/or a long enough training length, a linear inverse should in theory be able to suppress the out-of-band noise as well as the NRBFN inverse, and therefore perform as well as the NRBFN inverse. However, given the same number of kernels in a NRBFN inverse as taps in a linear inverse, as well as the same training length for each inverse,

it was observed that a NRBFN inverse could perform substantially better than a linear inverse.

8.3.4 Broomhead's filtering method using a bandstop linear IIR filter

A 6^{th} order bandstop IIR Butterworth filter, with a stopband of $f/f_s = 0.28$ to $f/f_s = 0.2825$, was used as the linear bandstop filter in Broomhead's filter method.

The PSD and the attractor plot of the IIR filtered noise corrupted sine wave signal, are given in Figure 8.11.



(a) **(b)**

Figure 8.11: *IIR filtered Ikeda noise corrupted sine wave, (a) PSD estimated using 2048 samples, (b) attractor plot.*

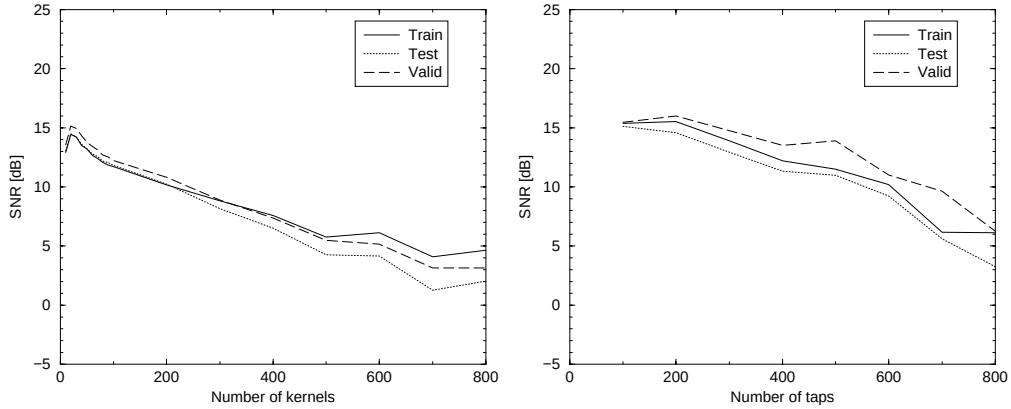
The removal of the sine wave by the IIR bandstop filter is evident from Figure 8.11(a). The filtered attractor in Figure 8.11(b) bears some resemblance to the original attractor in Figure 8.2(a), but is heavily distorted, or blurred.

Training the inverse with the signal of interest present

A NRBFN with an embedding dimension of 4 and a training length of 2500 samples was used as the nonlinear inverse. To test the generalisation properties of the nonlinear inverse, testing and validation data sets of length 2500 samples were used. The number of Gaussian kernels used in the NRBFN was varied from 10 to 90, in steps of 10, and from 100 to 800, in steps of 100. A linear comparison was used with a training length of 2500 samples. As with the nonlinear inverse, testing and validation data sets of length 2500 samples were used to judge the generalisation properties of the linear inverse. The number of taps in the linear inverse was

varied from 100 to 800, in steps of 100.

The achieved output SNR's for Broomhead's nonlinear inverse filter method, and for a linear inverse comparison are given in Figure 8.12.



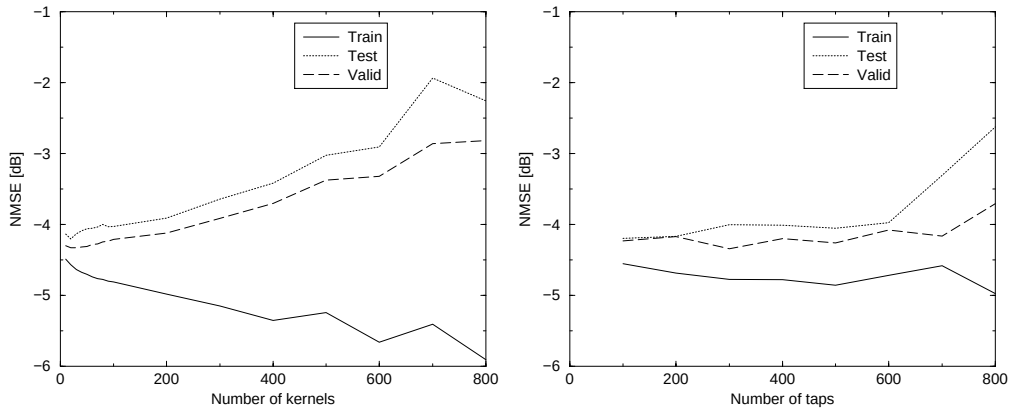
(a) **(b)**

Figure 8.12: *Output SNR's for (a) nonlinear inverse, and (b) linear inverse.*

As can be seen from Figure 8.12, the linear inverse was able to perform better than the nonlinear inverse. However, both inverses performed more poorly than the Butterworth filters mentioned in section 8.3.2. The evidence in Figure 8.12 suggests that when the noise process is not known *a priori*, and the inverse must be trained on the noise corrupted signal of interest data, the problem is completely, or mostly linear, and a linear inverse would be preferred to a nonlinear inverse. Also note from Figure 8.12, that with increasing complexity the linear and nonlinear inverses achieve poorer output SNR's. This is due to the fact that in the training phase, with ever more complexity, the inverse over-fits to the training data, which results in the noise corrupted signal of interest being cancelled, instead of only cancelling the noise. The NMSE results corresponding to the SNR results in Figure 8.12 are given in Figure 8.13.

Note that the results in Figures 8.12 and 8.13, are comparable with, but slightly poorer than those shown in Figures 8.5 and 8.6, for the notch bandstop linear filter case. It is suggested that the poorer performance of the inverses on the IIR filtered data could be due, at least in part, to the distortions, or blurring, introduced by the IIR filter, which were briefly discussed above.

The PSD of the recovered signal of interest, using a nonlinear inverse with an embedding dimension of 4, 100 kernels and a training length of 2500 samples, is shown in Figure 8.14(a).



(a) **(b)**

Figure 8.13: *NMSE's for (a) nonlinear inverse, (b) linear inverse.*

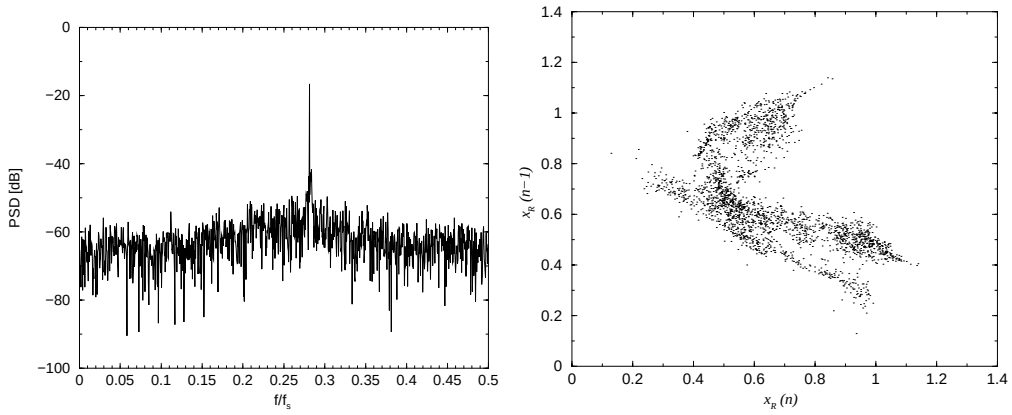
The attractor of the reconstructed Ikeda noise obtained using the same nonlinear inverse is shown in Figure 8.14(b). The results in Figure 8.14 were obtained using the testing data set.

The reconstructed attractor of Figure 8.14(b) resembles a blurred version of the reconstructed attractor given in Figure 8.7(b), which was obtained for the notch bandstop filter case, where the signal of interest was present during the training of the inverse.

Training the inverse without the signal of interest present

A NRBFN with an embedding dimension of 4 and a training length of 2500 samples was used as the nonlinear inverse. To test the generalisation properties of the nonlinear inverse, testing and validation data sets of length 2500 samples were used. The number of Gaussian kernels used in the NRBFN was varied from 100 to 800, in steps of 100. A linear comparison was used with a training length of 2500 samples. As with the nonlinear inverse, testing and validation data sets of length 2500 samples were used to judge the generalisation properties of the linear inverse. The number of taps in the linear inverse was varied from 100 to 800, in steps of 100.

The achieved output SNR's for Broomhead's nonlinear inverse filter method, and for a linear inverse filter are given in Figure 8.15. As can be seen from Figure 8.15, the performance of the nonlinear and linear inverses was better than for the case where the signal of interest was present during training. However, the performance was still below that of the IIR Butterworth bandpass filters, discussed in section 8.3.2. Furthermore, the performance of the nonlinear inverse was substantially poorer than that of the nonlinear inverse which was trained without the signal of



(a) **(b)**

Figure 8.14: *Testing data set results for Broomhead's filter method, using a NRBFN inverse with an embedding dimension of 4, 100 kernels, and a training length of 2500 samples: (a) PSD of the recovered signal of interest* $\{\hat{t}(n)\}$ *and (b) the attractor of the reconstructed Ikeda noise data.*

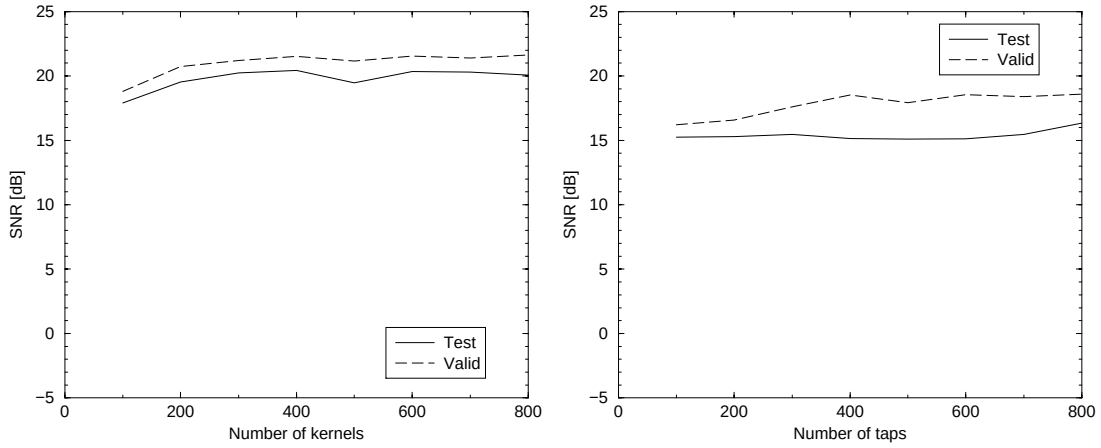
interest present when a notch filter was used to remove the signal of interest, see Figure 8.8(a). The NMSE's corresponding to the SNR's of Figure 8.15 are given in Figure 8.16.

The PSD of the recovered signal of interest, using a nonlinear inverse with an embedding dimension of 4, 800 kernels and a training length of 2500 samples, is shown in Figure 8.17(a). The attractor of the reconstructed Ikeda noise obtained using the same nonlinear inverse is shown in Figure 8.17(b). The results in Figure 8.17 were obtained using the testing data set.

The reconstructed attractor of Figure 8.17(b) resembles a blurred version of the reconstructed attractor given in Figure 8.10(b), which was obtained for the notch bandstop filter case, where the signal of interest was not present during the training of the inverse.

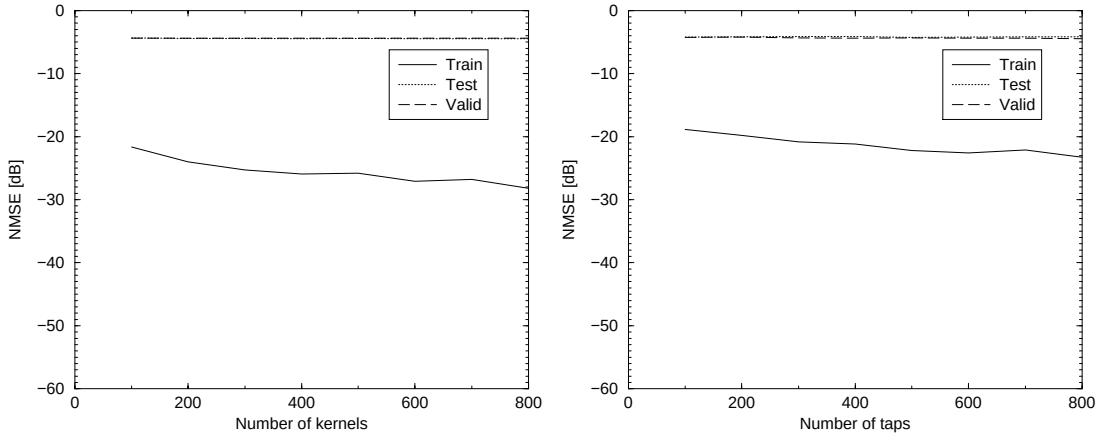
8.3.5 Broomhead's filtering method using a bandstop FIR filter

Broomhead's filter method was applied to the sine wave in Ikeda noise problem, using a (non-notch) conventional linear FIR bandstop filter [127]. This was done to investigate if such a filter could be used to preserve the dynamics (or more generally the nonlinear properties) of the Ikeda noise process. As mentioned in section 8.2.2, there is a recognised trade-off between preservation of dynamics and signal suppression, in terms of choosing a suitable filter length, or order. For this reason two different lengths of bandstop FIR filter were used. FIR filters with



(a) **(b)**

Figure 8.15: *Output SNR's for (a) nonlinear inverse, (b) linear inverse.*



(a) **(b)**

Figure 8.16: *NMSE's for (a) nonlinear inverse, (b) linear inverse.*

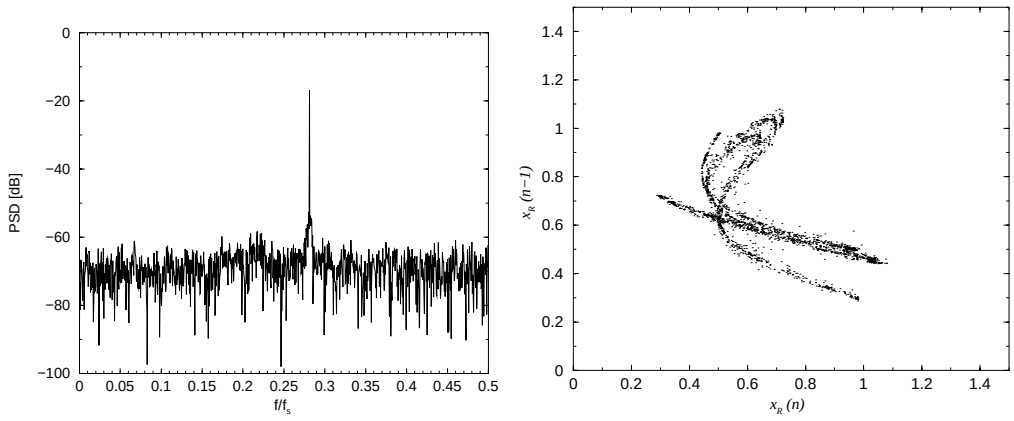


Figure 8.17: *Testing data set results for Broomhead's filter method, using a NRBFN inverse with an embedding dimension of 4, 800 kernels, and a training length of 2500 samples: (a) PSD of the recovered signal of interest* $\{\hat{t}(n)\}$ *and (b) the attractor of the reconstructed Ikeda noise data.*

25 taps, and 193 taps were considered. Moreover, in general, the signal of interest is likely to be more complicated than a simple sinusoid, and would occupy a larger bandwidth. Therefore, to get an idea of how FIR filters would alter the nonlinear properties of a signal, when the signal of interest occupied a narrow frequency band (rather than just one frequency which is the case for a sinusoid), the FIR filters were chosen to suppress a frequency band of $f/f_s = 0.26$, up to $f/f_s = 0.30$. PSD and attractor plots are given in Figure 8.18, of the bandstop FIR filtered data. As can be seen in Figure 8.18, both of the FIR filters failed to completely suppress the signal of interest. The attractor plots of the filtered data are very distorted, and blurred, with respect to the attractor of the original Ikeda noise attractor, see Figure 8.2(a).

The nonlinear and linear experiments carried out for the notch and IIR bandstop filters, in sections 8.3.3 and 8.3.4, respectively, were repeated for both the 25 tap FIR bandstop filter, and for the 193 tap FIR bandstop filter. A NRBFN with an embedding dimension of 4 and a training length of 2500 samples was used as the nonlinear inverse. The number of Gaussian kernels used in the NRBFN was varied from 100 to 800, in steps of 100. A linear comparison was used with a training length of 2500 samples. The number of taps in the linear inverse was varied from 100 to 800, in steps of 100. As in sections 8.3.3 and 8.3.4, the training of the nonlinear and linear inverses was considered for the case when the signal of interest was present during training, and for the case when it was not. For both cases it was found that the nonlinear and linear

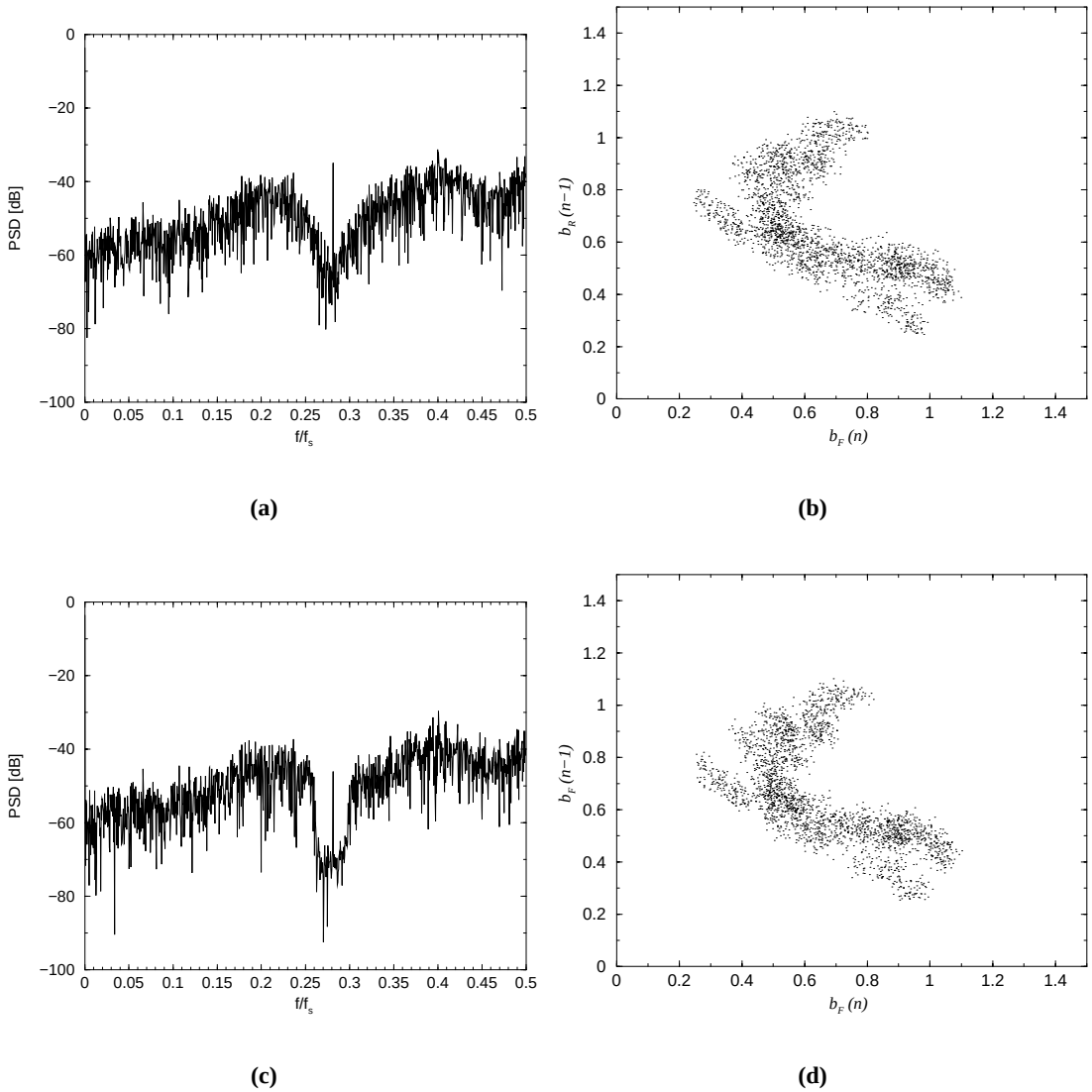


Figure 8.18: *PSD and attractor plots for FIR filtered sine plus Ikeda data: (a) PSD and (b) attractor plots for data filtered using 25 tap FIR, (c) PSD and (d) attractor plots for data filtered using 193 tap FIR.*

inverses performed very poorly, producing very little improvement in the SNR, or even worse, degrading the SNR. SNR results are presented for the case where the signal of interest was present during training of the inverses in Figure 8.19, and in Figure 8.20 for the case where the signal of interest was not present during the training of the inverses. The poor performance of

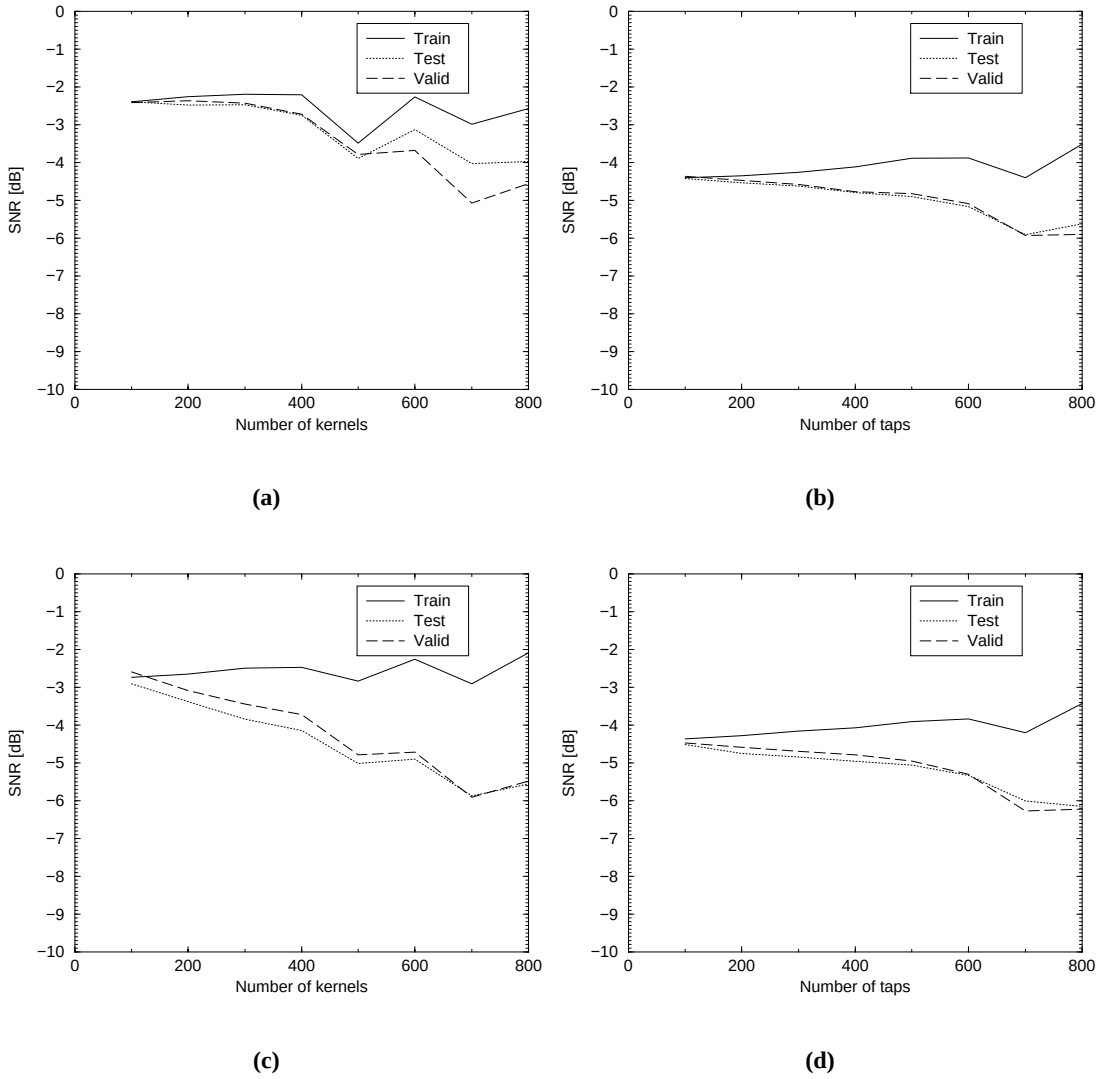


Figure 8.19: *SNR's for nonlinear and linear inverses with the signal of interest present during training, using FIR filter lengths of 25 and 193 taps: (a) nonlinear inverse and (b) linear inverse using an FIR with 25 taps, (c) nonlinear inverse and (d) linear inverse using an FIR with 193 taps.*

the nonlinear and linear inverses was attributed to both the blurring of the attractor caused by the FIR filters, and also to the bandstop filters' incomplete removal of the sine wave signal of interest.

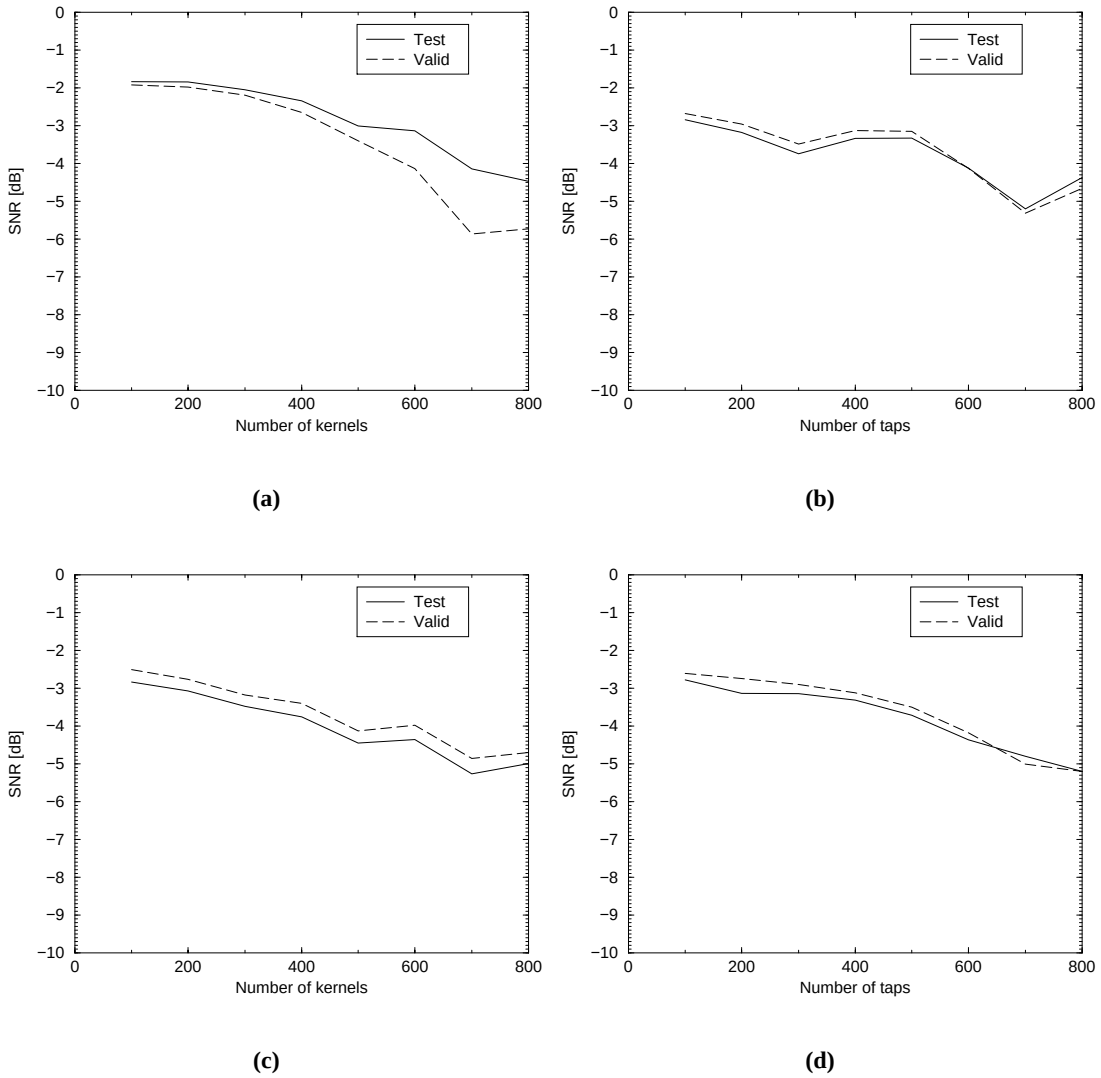


Figure 8.20: *SNR's for nonlinear and linear inverses without the signal of interest present during training, using FIR filter lengths of 25 and 193 taps: (a) nonlinear inverse and (b) linear inverse using an FIR with 25 taps, (c) nonlinear inverse and (d) linear inverse using an FIR with 193 taps.*

8.4 Broomhead's filter method with a novel linear bandstop filtering approach

8.4.1 Introduction

From the results in section 8.3, it would appear that the only scenario where Broomhead's nonlinear inverse filtering method would be preferred to a conventional linear alternative, would be when the signal of interest could be suppressed by a notch filter, and the noise process was known *a priori*, so that it could be used to train the nonlinear inverse. In section 8.3.3 it was demonstrated that a notch filter was able to both suppress the signal of interest, and preserve the dynamics of the Ikeda noise process, which allowed a nonlinear inverse to outperform both a linear inverse, and a linear Butterworth IIR bandpass filter. However, in general, the signal of interest will not be a simple sinusoid, and there would be a need to knock out several frequencies, not just one.

The novel linear bandstop filtering approach, examined in sections 8.4.3 and 8.4.4, is to use a filter with several notches (*i.e.* several zeros on the unit circle), which sufficiently suppresses the signal of interest, but also preserves the dynamics, or nonlinear properties, of the noise process concerned. To consider whether or not this is a practical idea, Gaussian narrowband signals were added to the Ikeda noise process discussed in section 8.3.1. The narrowband signals are discussed in section 8.4.2, and the results are reported for 2 SNR cases, 8.5dB and -8.5dB, in sections 8.4.3 and 8.4.4.

8.4.2 Narrowband signals

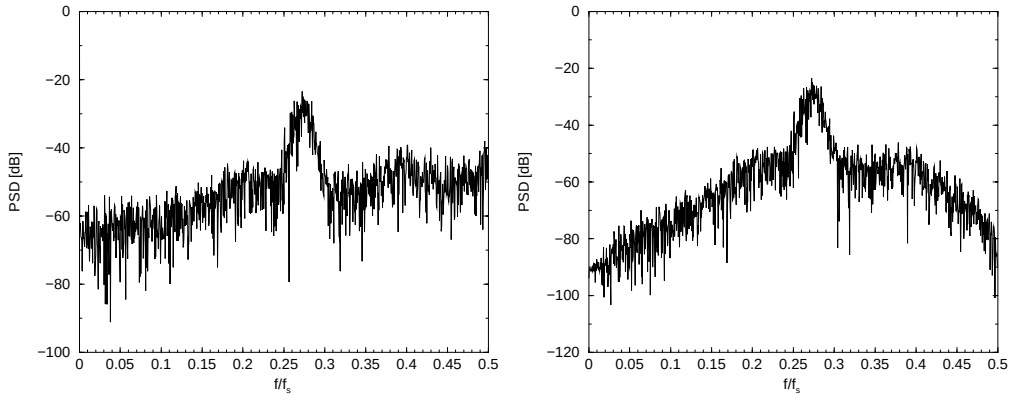
The narrowband signals were generated by filtering Gaussian white noise with an IIR Butterworth bandpass filter. Table 8.3 gives the parameters of the filtered Gaussian noise processes, along with the IIR filter parameters used to obtain the narrowband Gaussian signals. The (narrowband signal to Ikeda noise) SNR is also given.

Gaussian variance	**Gaussian mean**	**IIR filter order**	**IIR filter passband**	**SNR [dB]**
5	0	6^{th}	$f/f_s = 0.26 \rightarrow f/f_s = 0.285$	8.5
0.1	0	6^{th}	$f/f_s = 0.26 \rightarrow f/f_s = 0.285$	-8.5

Table 8.3: *Generation of the Gaussian narrowband signals of interest.*

8.4.3 Experiment 1: an input SNR of 8.5dB

A benchmark performance measure for the linear and nonlinear inverses was established using an IIR Butterworth bandpass filter. It was found that using 6^{th} and 4^{th} order filters altered the narrowband signal of interest, therefore a 2^{nd} order filter was used. A trial and error approach was adopted to find a passband range that did not affect the narrowband signal of interest. A passband of $f/f_s = 0.22$ to$f/f_s = 0.34$ was found to both not affect the signal of interest, and to slightly improve the SNR. The output SNR using this Butterworth bandpass filter was 11.4dB. The PSD of the Gaussian narrowband signal corrupted by Ikeda noise is given in Figure 8.21(a). The PSD of the bandpass filtered Gaussian narrowband signal corrupted by Ikeda noise, is given in Figure 8.21(b).



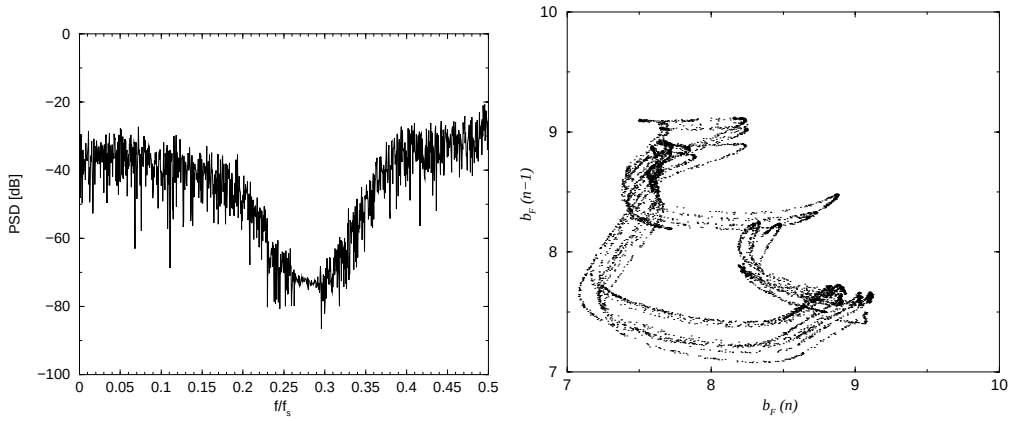
(a) **(b)**

Figure 8.21: *PSD of (a) Gaussian narrowband signal corrupted by Ikeda noise, and (b) PSD of bandpass filtered narrowband signal corrupted by Ikeda noise.*

To suppress the narrowband signal three single notch filters were used in cascade[3]. The gain G of each filter was 1.0, and the notches were placed at the following frequencies, $f/f_s = 0.265, f/f_s = 0.275, f/f_s = 0.285$. The PSD and attractor plots of the notch filtered data are given in Figure 8.22.

Note the similarity (despite a few additional twists and turns) between the filtered attractor plot of Figure 8.22(b), and the filtered attractor plot of the single notch filtered data, given in Figure 8.4(b).

[3] A single filter with three notches could equally have been used.



(a) **(b)**

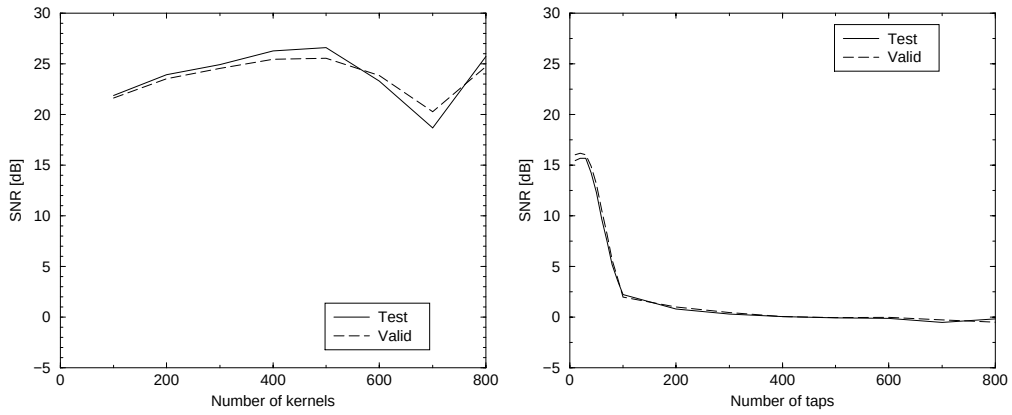
Figure 8.22: *PSD of (a) notch filtered data, and (b) the attractor of the notch filtered data.*

A NRBFN with an embedding dimension of 4 and a training length of 2500 samples was used as the nonlinear inverse in Broomhead's filter method. To test the generalisation properties of the nonlinear inverse, testing and validation data sets of length 2500 samples were used. The number of Gaussian kernels used in the NRBFN was varied from 100 to 800, in steps of 100. A linear comparison was used, with a training length of 2500 samples. As with the nonlinear inverse, testing and validation data sets of length 2500 samples were used to judge the generalisation properties of the linear inverse. The number of taps was varied from 10 to 90 taps, in steps of 10, and from 100 to 800 taps in steps of 100. The training of the nonlinear and linear inverses was done, for all simulations, using noise only data (*i.e.* not the noise corrupted signal of interest). The output SNR results obtained by the nonlinear and linear inverses are shown in Figure 8.23.

As can be seen from Figure 8.23, both the linear and nonlinear inverses managed to achieve a better output SNR than the linear 2^{nd} order Butterworth bandpass filter. The nonlinear inverse achieved better output SNR values than the linear inverse.

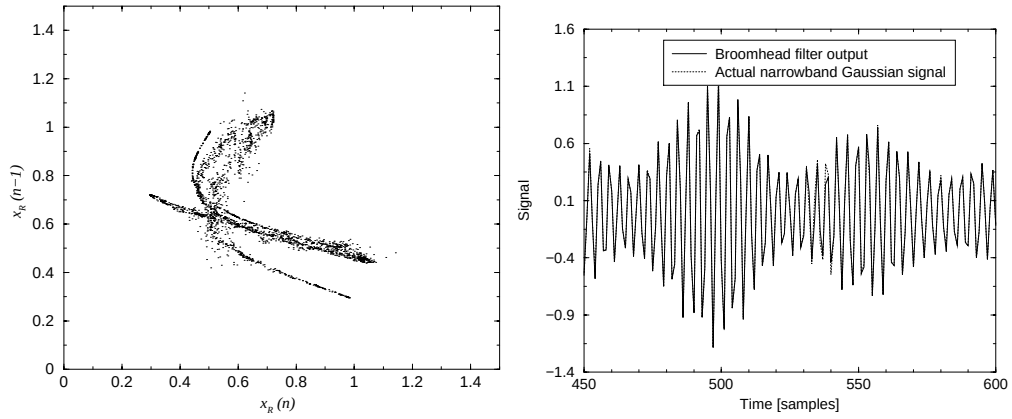
The attractor of the reconstructed Ikeda noise process, and a plot of the output of Broomhead's filter method, for the testing data set are given in Figure 8.24.

As can be seen from Figure 8.24(a), the reconstructed attractor resembles the attractor of the original Ikeda noise process, see Figure 8.2(a), despite discernible distortions, or blurring.



(a) **(b)**

Figure 8.23: *Output SNR's for (a) the nonlinear inverse, and (b) a linear inverse comparison.*

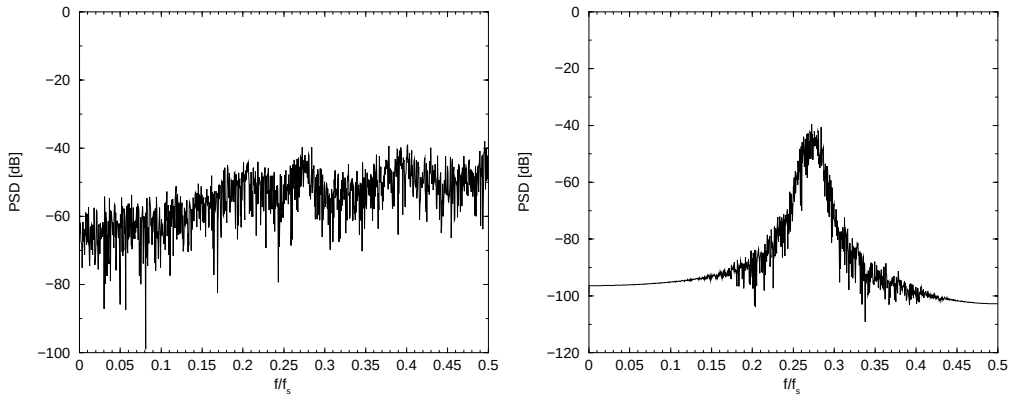


(a) **(b)**

Figure 8.24: *Testing data set results obtained using Broomhead's (nonlinear inverse) filter method, for the case when 500 kernels were used: (a) the reconstructed Ikeda noise attractor, and (b) a plot of the Broomhead filter output versus the signal of interest.*

8.4.4 Experiment 2: an input SNR of -8.5dB

A benchmark performance measure for the linear and nonlinear inverses was established using an IIR Butterworth bandpass filter. A 6^{th} order filter with a passband of $f/f_s = 0.258$ to $f/f_s = 0.287$ was used. The output SNR using this Butterworth bandpass filter was -1dB. The PSD of the Gaussian narrowband signal corrupted by Ikeda noise is given in Figure 8.25(a). The PSD of the bandpass filtered Gaussian narrowband signal corrupted by Ikeda noise, is given in Figure 8.25(b).

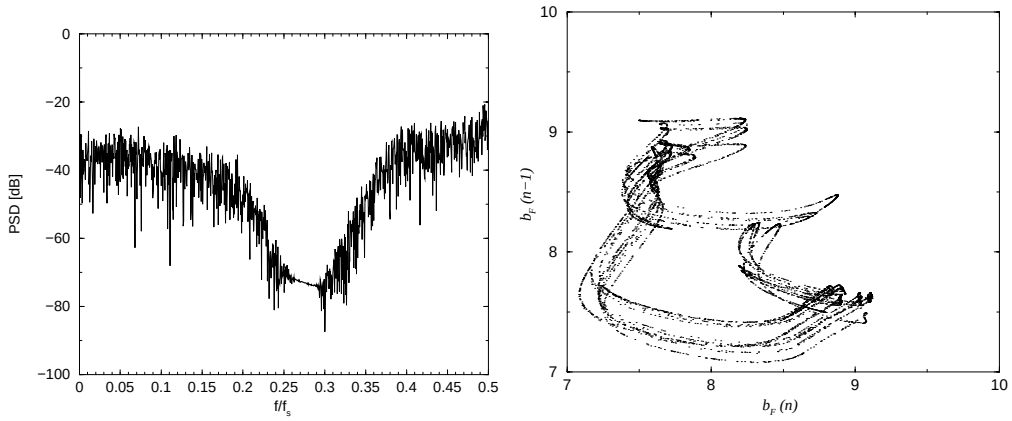


(a) **(b)**

Figure 8.25: *PSD of (a) Gaussian narrowband signal corrupted by Ikeda noise, and (b) PSD of bandpass filtered narrowband signal.*

To suppress the narrowband signal three single notch filters were used in cascade. The gain G of each filter was 1.0, and the notches were placed at the following frequencies, $f/f_s = 0.265, f/f_s = 0.275, f/f_s = 0.285$. The PSD and attractor plots of the notch filtered data are given in Figure 8.26.

A NRBFN with an embedding dimension of 4 and a training length of 2500 samples was used as the nonlinear inverse in Broomhead's filter method. To test the generalisation properties of the nonlinear inverse, testing and validation data sets of length 2500 samples were used. The number of Gaussian kernels used in the NRBFN was varied from 100 to 800, in steps of 100. A linear comparison was used, with a training length of 2500 samples. As with the nonlinear inverse, testing and validation data sets of length 2500 samples were used to judge the generalisation properties of the linear inverse. The number of taps was varied from 10 to 90 taps, in steps of 10, and from 100 to 800 taps in steps of 100. The training of the nonlinear and

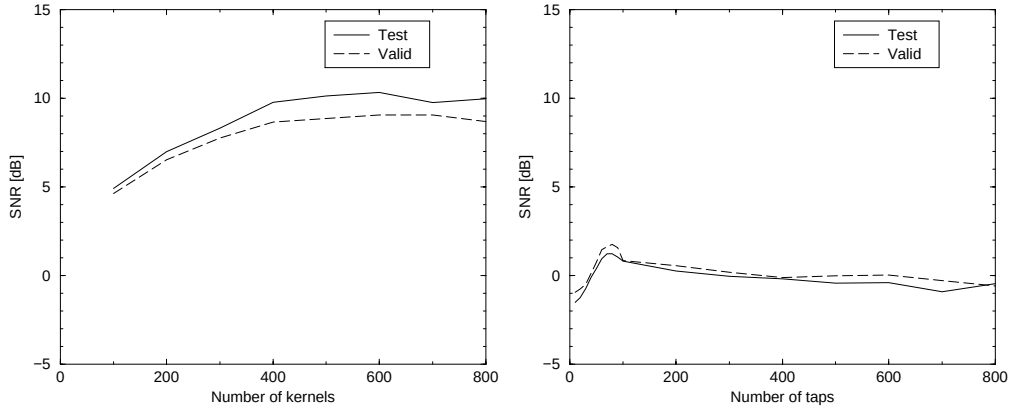


(a) **(b)**

Figure 8.26: *PSD of (a) notch filtered data, and (b) the attractor of the notch filtered data.*

linear inverses was done, for all simulations, using noise only data (*i.e.* not the noise corrupted signal of interest).

The output SNR results obtained by the nonlinear and linear inverses are shown in Figure 8.27.

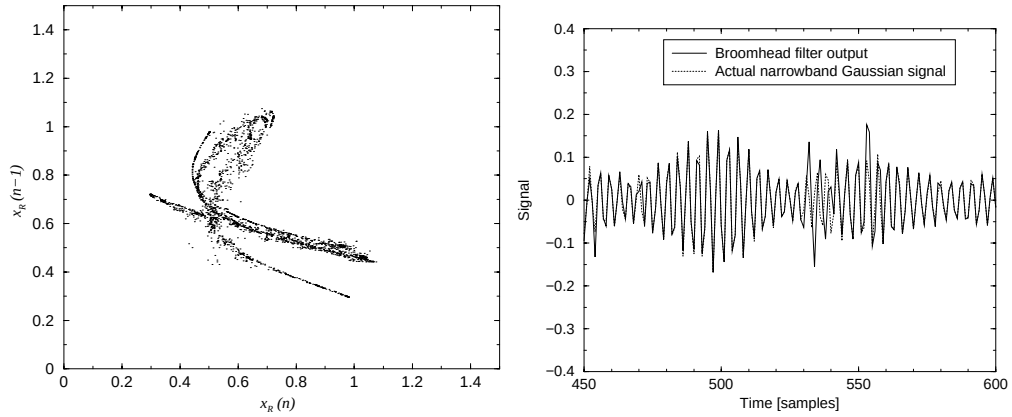


(a) **(b)**

Figure 8.27: *Output SNR's for (a) the nonlinear inverse, and (b) a linear inverse comparison.*

As can be seen from Figure 8.27, both the linear and nonlinear inverses managed to achieve a better output SNR than the linear 6^{th} order Butterworth bandpass filter. The nonlinear inverse achieved better output SNR values than the linear inverse. Note that a similar improvement in SNR was obtained using the nonlinear inverse for this experiment, as was obtained using a nonlinear inverse in experiment 1: the best improvement in SNR was, approximately, 18dB in

both cases. The attractor of the reconstructed Ikeda noise process, and a plot of the output of Broomhead's filter method, for the testing data set are given in Figure 8.28.



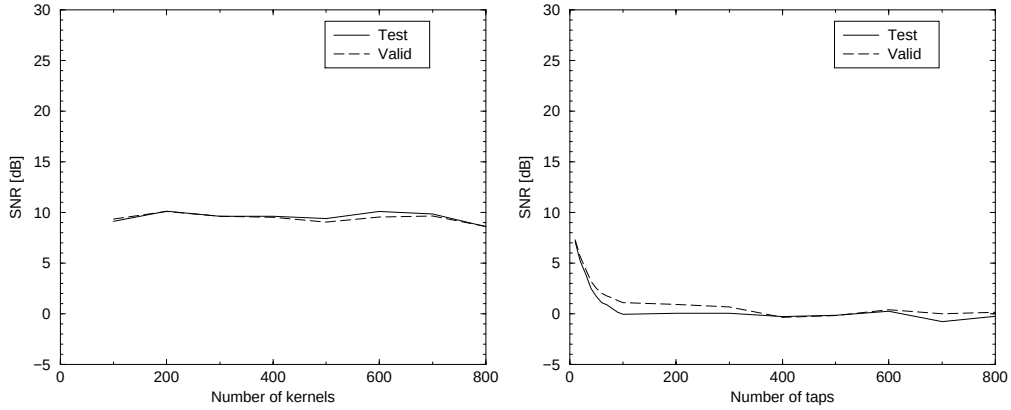
(a) **(b)**

Figure 8.28: *Testing data set results obtained using Broomhead's (nonlinear inverse) filter method, for the case when 600 kernels were used: (a) the reconstructed Ikeda noise attractor, and (b) a plot of the Broomhead filter output versus the signal of interest.*

As can be seen from Figure 8.28(a), the reconstructed attractor resembles the attractor of the original Ikeda noise process, see Figure 8.2(a), despite discernible distortions, or blurring.

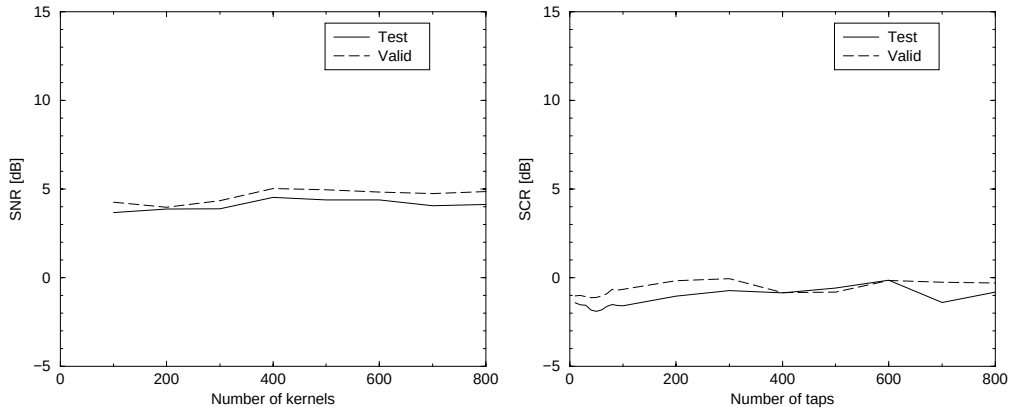
8.4.5 Using a linear IIR bandstop filter

The Broomhead filter method, with an IIR bandstop filter, was used as a comparison with Broomhead's filter method which used the cascade of notch filters, to cancel the Gaussian narrowband signals of interest. The nonlinear and linear inverse experiments carried out in sections 8.4.3 and 8.4.4, were repeated using an IIR bandstop filter, instead of a cascade of notch filters. A 6^{th} order Butterworth bandstop filter, with a stopband from $f/f_s = 0.258$ to $f/f_s = 0.287$ was used. The results for the nonlinear and linear inverses are given in Figure 8.29 for an input SNR of 8.5dB, and in Figure 8.30 for an input SNR of -8.5dB. As can be seen from Figures 8.29 and 8.30, the Broomhead filter results obtained using an IIR bandstop filter were poorer than when a cascade of notch filters was used. This is because the IIR filter distorted the dynamics of the Ikeda noise process more than the cascade of notch filters did. This subject has already been discussed in section 8.3.



(a) **(b)**

Figure 8.29: *Output SNR's for (a) the nonlinear inverse, and (b) a linear inverse comparison, for an input SNR of 8.5dB.*



(a) **(b)**

Figure 8.30: *Output SNR's for (a) the nonlinear inverse, and (b) a linear inverse comparison, for an input SNR of -8.5dB.*

8.5 Broomhead's filter method applied to the cancellation of sea clutter

8.5.1 Introduction

The results of section 8.4, demonstrate that the novel bandstop linear filtering approach, of using a number of notches to suppress a narrowband signal, can lead to performance improvements, relative to linear alternatives, when incorporated into Broomhead's nonlinear inverse filter method. With these promising results in mind, the approach was applied to the cancellation of sea clutter. The motivation behind using this approach was, that if Broomhead's filter method could be used to improve the signal to clutter ratio, then this could be translated into a detection performance improvement.

To consider the cancellation of sea clutter using Broomhead's filter method, two sea clutter data sets were picked. An example from the wavetank data sets was chosen: the $12ms^{-1}$ gate 14 amplitude data set. A Dawber data set was also used: the amplitude VV data set. Considering the PSD of these two signals, it was decided to use narrowband signals with a start band frequency of $f/f_s = 0.0$. See Figures 8.31, and 6.1(a) for PSD plots for the wavetank gate 14 and Dawber VV amplitude data sets, respectively.

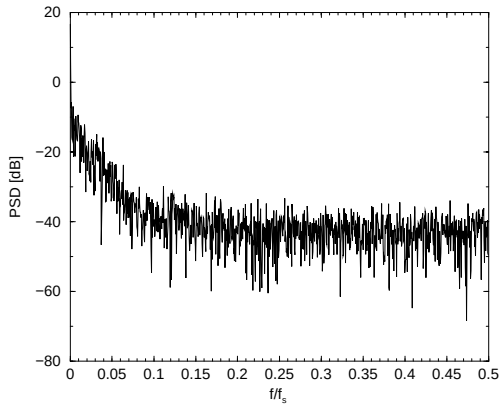


Figure 8.31: *PSD of* $12ms^{-1}$ *gate 14 wavetank amplitude data set.*

A description of the narrowband signals, used for the investigation into the cancellation of the two sea clutter data sets mentioned above, is given in section 8.5.2. The results for the cancellation of the wavetank data set are given in section 8.5.3. The results for the cancellation of the Dawber VV data set are given in section 8.5.4.

8.5.2 Narrowband signals

The narrowband signals were generated by filtering Gaussian white noise with an IIR Butterworth lowpass filter. Table 8.4 gives the narrowband Gaussian signal generation parameters and resulting SCR's for the cancellation of the wavetank data set experiment. Table 8.5 shows the narrowband signal generation parameters and resulting SCR's for the cancellation of the Dawber data set experiment.

Gaussian variance	**Gaussian mean**	**IIR filter order**	**IIR filter passband**	**SCR [dB]**
10,000	0	6^{th}	$f/f_s = 0.0 \rightarrow f/f_s = 0.025$	7.3
1000	0	6^{th}	$f/f_s = 0.0 \rightarrow f/f_s = 0.025$	-2.7

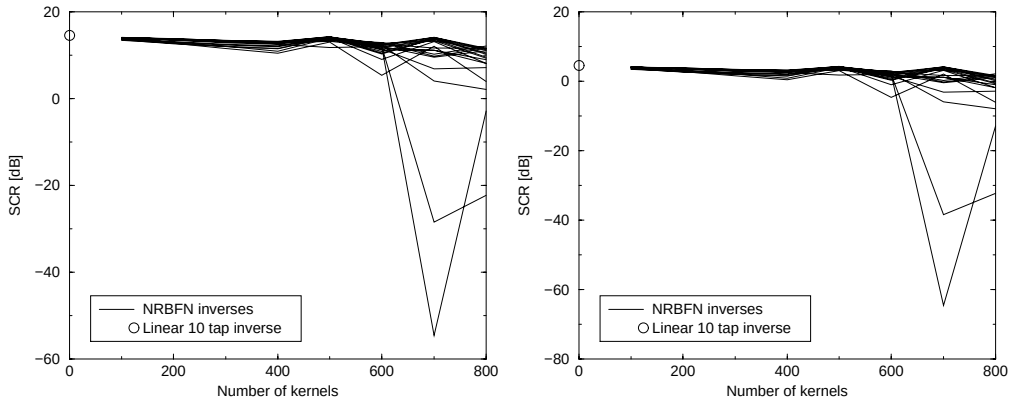
Table 8.4: *Generation of the Gaussian narrowband signals of interest for the cancellation of the* $12ms^{-1}$ *gate 14 amplitude data set .*

Gaussian variance	**Gaussian mean**	**IIR filter order**	**IIR filter passband**	**SCR [dB]**
0.001	0	6^{th}	$f/f_s = 0.0 \rightarrow f/f_s = 0.025$	7.9
0.0001	0	6^{th}	$f/f_s = 0.0 \rightarrow f/f_s = 0.025$	-2.1

Table 8.5: *Generation of the Gaussian narrowband signals of interest for the cancellation of the Dawber VV amplitude data set .*

8.5.3 Cancellation of the wavetank data set

The following simulation parameters were used to cancel the narrowband signals of interest given in Table 8.4, for both the nonlinear and linear inverses. Three single notch filters were used in cascade. The gain G of each filter was 1.0, and the notches were placed at the following frequencies, $f/f_s = 0.005, f/f_s = 0.015,$ and $f/f_s = 0.025$. A NRBFN was used as the nonlinear inverse in Broomhead's filter method. Embedding dimensions of 4 to 20 in steps of 1, and of 30, 40, and 50 were used. For each embedding dimension, the number of kernels was varied from 100 to 800 in steps of 100. The training length for each simulation was 2500 samples. A 10 tap linear inverse comparison was used. The training of the nonlinear and linear inverses was done, for all simulations, using clutter only data (*i.e.* not the clutter corrupted signal of interest). Testing data set results for the linear and nonlinear inverses are given in Figure 8.32.



(a) **(b)**

Figure 8.32: *Output SCR's for the nonlinear and linear inverses for (a) an input SCR of 7.3dB, and (b) an input SCR of -2.7dB.*

As can be seen from Figures 8.32(a) and 8.32(b), for all the NRBFN inverses considered, the 10 tap linear inverse performed as well as or better than each NRBFN inverse. This provides convincing evidence that Broomhead's nonlinear inverse filter method cannot be used to improve the output SCR, with respect to what can be achieved using linear alternatives, for the $12ms^{-1}$ gate 14 amplitude data set.

8.5.4 Cancellation of the Dawber data set

The following simulation parameters were used to cancel the narrowband signals of interest given in Table 8.5, for both the nonlinear and linear inverses. Three single notch filters were used in cascade. The gain G of each filter was 1.0, and the notches were placed at the following frequencies, $f/f_s = 0.005, f/f_s = 0.015,$ and $f/f_s = 0.025$. A NRBFN was used as the nonlinear inverse in Broomhead's filter method. Embedding dimensions of 4 to 20 in steps of 1, and of 30, 40, and 50 were used. For each embedding dimension, the number of kernels was varied from 100 to 800 in steps of 100. The training length for each simulation was 2500 samples. A 10 tap linear inverse comparison was used. The training of the nonlinear and linear inverses was done, for all simulations, using clutter only data (*i.e.* not the clutter corrupted signal of interest). Testing data set results for the linear and nonlinear inverses are given in Figure 8.33.

As can be seen from Figures 8.33(a) and 8.33(b), for all NRBFN inverses considered, the 10

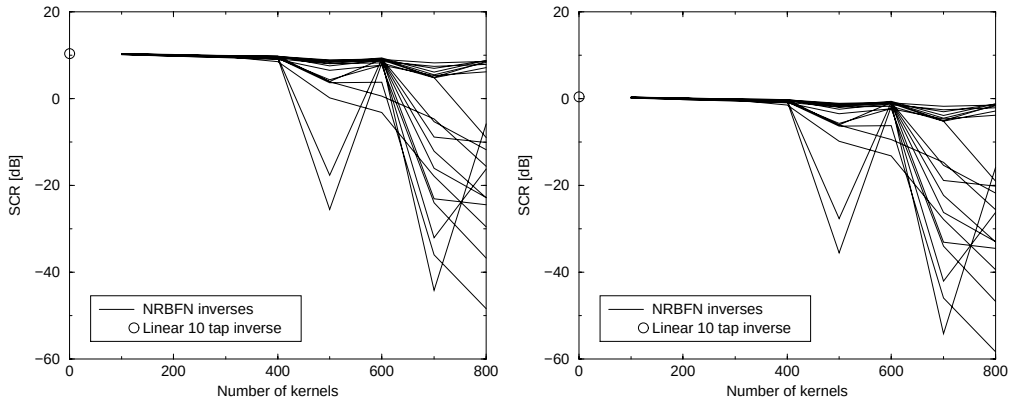


Figure 8.33: *Output SCR's for the nonlinear and linear inverses for (a) an input SCR of 7.9dB, and (b) an input SCR of -2.1dB.*

tap linear inverse performed as well as or better than the NRBFN inverse. This provides convincing evidence that Broomhead's nonlinear inverse filter method cannot be used to improve the output SCR, with respect to what can be achieved using linear alternatives, for the Dawber VV amplitude data set.

8.6 Summary

Broomhead's nonlinear inverse filter method was applied to the task of cancelling chaotic noise from a sine wave, using a number of different linear bandstop filtering approaches. Improvements were observed, with respect to linear alternatives, if a notch filter was used as the bandstop linear filter in Broomhead's nonlinear inverse filter method, but not for any of the other bandstop filters considered. However, these improvements, relative to linear alternatives, were only obtained if it was possible to train the nonlinear inverse on noise only data. The notch filter was able to both adequately remove the signal of interest (*i.e.* a sine wave), as well as limit the distortion to the dynamics of the chaotic noise process. The results from this simple sine wave experiment gave rise to a novel bandstop filtering idea, for use in Broomhead's filter method. This novel approach was intended for cases when the signal of interest was more complicated than a simple sine wave, and had more than one frequency component. The novel approach involved using several notches (or zeros on the unit circle) to adequately suppress the signal of interest, but which also preserved the nonlinear properties of the nonlinear (possibly chaotic)

noise process. This approach was shown to be successful when using a narrowband Gaussian signal of interest corrupted by broadband Ikeda noise: Broomhead's nonlinear inverse filter method achieved better output SCR's than were obtained by linear alternatives. This approach was also applied to two sea clutter data sets. It was found that for both sea clutter data sets, a linear inverse technique performed as well as, or better than, a nonlinear inverse. Therefore Broomhead's nonlinear inverse filter method was not preferred to linear alternatives for these clutter data sets. Due to the similar properties of all the sea clutter data sets, as highlighted in Chapter 5 by the prediction analysis carried out, it was inferred that Broomhead's nonlinear inverse filter method would not be preferred to linear alternatives, for the cancellation of any of the sea clutter data sets described in Appendix A.

Chapter 9
Summary, conclusions, and suggestions for further research

9.1 Summary

The purpose of this thesis was to investigate the suitability of using nonlinear signal processing techniques in situations where non-Gaussian stochastic or chaotic deterministic time series are encountered. The specific application of interest for this thesis has been maritime surveillance radar. The salient points from this thesis are now briefly summarised.

In Chapter 2 the modelling of sea clutter was discussed. Key aspects of chaos theory were introduced which laid the foundations upon which the claim, made by Haykin and his co-workers, that sea clutter is chaotic could be reviewed and assessed. The most important fact to be highlighted in this review was that chaotic invariants cannot be used to distinguish between chaotic and stochastic processes. The traditional stochastic modelling of sea clutter was also described in this chapter. In particular, attention was drawn to the popular compound K-distribution model. This model not only has some physical justification for its use, but it also allows for a realistic treatment of the correlation properties of sea clutter.

In Chapter 3 a technique was presented for the estimation of the compound K-distributed components of a sea clutter data set. A simulation technique for the generation of surrogate correlated compound K-distributed sea clutter was described.

In Chapter 4 some chaos theory ideas, which were introduced in Chapter 2, were used in the construction of nonlinear predictors for chaotic signals. Two nonlinear models were used for these predictors: the VSF, and the RBFN. Global linear and nonlinear prediction analysis was carried out on two chaotic time series: from the Logistic map and the Lorenz attractor. Nonlinear predictors were shown to perform better than linear predictors, in terms of NMSE, on these chaotic time series. In particular, a NRBFNP was shown to perform better than an UNRBFNP, and for a given RBFNP there was observed to be little performance difference between using the RSC method which picked kernel centres at random from a training data set, and

the OAKM method which used a clustering algorithm to pick kernel centres from a training data set. It was found that the embedding delay which resulted in the best NMSE prediction estimate for a RBFNP was 1 sample, for both Logistic map and Lorenz data. Furthermore, a NRBFNP was shown to be capable of capturing the underlying dynamics of a chaotic signal corrupted by white Gaussian noise, whereas attempts using an UNRBFNP, and an UNRBFNP with regularisation failed on the same data.

In Chapter 5 a comprehensive prediction analysis was reported for the wavetank and Dawber sea clutter data sets described in Appendix A. Global linear and nonlinear prediction results provided evidence to suggest that the wavetank and Dawber data sets have linear predictor functions. Furthermore, recursive prediction revealed a lack of evidence of underlying nonlinear chaotic dynamics for these data sets. Indeed, rather than the chaotic model proposed by Haykin and his co-workers, a linear compound Gaussian stochastic model was shown to be suitable for the wavetank and Dawber clutter data sets.

In Chapter 6 it was investigated if noise was preventing the detection of nonlinear predictability and chaotic behaviour in the wavetank and Dawber data sets. It was also investigated whether or not forward-backward prediction could be used to uncover nonlinearities not revealed using forward prediction alone. The results of these investigations merely provided further evidence to suggest that the wavetank and Dawber data sets have linear predictor functions. Also in this chapter, it was observed that substantial prediction improvements, in terms of NMSE, were achievable using forward-backward prediction techniques on Lorenz data, with respect to that achievable using forward prediction alone.

In Chapter 7 the incorporation of NLP's into maritime surveillance radar detection algorithms was considered, using the wavetank and Dawber data sets. As would be expected from the results reported in Chapters 5 and 6, it was found that the incorporation of NLP's into such algorithms was not justifiable, as a LPD could achieve as good as, or better, performance than a NLPD. Additionally, the importance of capturing a chaotic signal's underlying dynamics was considered for a NLP which was to be incorporated into a detection algorithm. It was demonstrated that modelling the underlying dynamics could have a detrimental effect on the detection performance of a NLPD, in terms of a ROC performance measure.

In Chapter 8 Broomhead's nonlinear inverse noise cancellation technique was re-investigated using a sine wave corrupted by broadband chaotic noise. It was demonstrated that significant

improvements, relative to linear alternatives, could be obtained using Broomhead's method, countering recent work which suggested otherwise. For such improvements to be obtained, the nonlinear inverse had to have access to the chaotic noise during its training phase, and a notch filter had to be used as the *bandstop* linear filter. The restriction of having to use a notch filter as the *bandstop* filter was highlighted. Moreover, a novel bandstop filtering approach was applied to Broomhead's filtering method which was capable of circumventing this restriction. This modified Broomhead filtering technique was shown to cancel broadband chaotic noise from a narrowband Gaussian signal better than alternative linear methods. Unfortunately the modified Broomhead filtering approach was shown to only perform as well as, or more poorly than, a linear technique, for the cancellation of the wavetank and Dawber clutter data sets from narrowband Gaussian signals.

9.2 Conclusions

9.2.1 Conclusions related to the nonlinear processing of chaotic signals

A global NLP was demonstrated to perform better than a global LP on chaotic time series. This supports previously published results and it is concluded that the reason for this is due to the nonlinear evolution of a chaotic attractor in state space.

If using a RBFN for the nonlinear prediction of a chaotic signal, a NRBFNP should be used in preference to an UNRBFNP, or a regularised UNRBFNP. Additionally, it was noted that using a clustering algorithm to pick RBFN kernels did not seem to offer much improvement over selecting centres at random from a training data set. It was determined that the less computationally expensive random selection technique should be implemented, and would perform as well as a RBFN which used the clustering selection method, as long as a sufficiently large number of kernels was used, which could sufficiently span the input space of the RBFN. Early-stopping could be used to avoid over-fitting that might result due to a large number of kernels being used.

A NRBFNP should be used, in preference to a UNRBFNP and a regularised UNRBFNP, to capture and model underlying dynamics, since a NRBFNP was shown to capture the underlying dynamics of a chaotic signal corrupted by noise, whereas the other networks did not. A NRBFNP is simpler to implement than a regularised UNRBFNP, as it does not involve searching for a suitable regularisation parameter.

Forward-backward prediction can be used to obtain substantial performance improvements, with respect to forward prediction alone.

The modified Broomhead filtering method shows promise for the cancellation of broadband chaotic noise from narrowband signals of interest. Suggestions for further work in this area are presented in section 9.3.

9.2.2 Conclusions related to the nonlinear processing of sea clutter

A global LP was shown to perform as well as, or better than, a NLP on every amplitude, in-phase, and quadrature-phase clutter data set analysed, therefore it is concluded that the wavetank and Dawber data sets have linear predictor functions. No evidence of underlying dynamics was found in any of the clutter data sets analysed using recursive prediction. The reason for this was determined to be that the data sets did not have any underlying dynamics. Indeed, a compound Gaussian stochastic model was shown to be appropriate for both the wavetank and Dawber data sets. This information leads to the conclusion that the data sets analysed were not generated by a phenomenon which may be modelled by a coupled system of nonlinear differential equations, and that they are therefore not examples of chaotic time series. This conclusion contradicts the claim made by Haykin and his co-workers that sea clutter is a chaotic process. However, differences between the data sets analysed by Haykin and his co-workers and those in Appendix A were highlighted in Chapter 5, and it was stated that these differences could be responsible for the lack of evidence of chaotic behaviour. In the case of the wavetank data it was observed that the wavetank in which the data sets were collected might not accurately represent conditions at sea, and that pulse compression could mask or remove any evidence of chaotic behaviour. In the case of the Dawber data it was pointed out that the data records only lasted for a period of 1.28 seconds, whereas those analysed by Haykin and his co-workers lasted for between 25 and 50 seconds. It was speculated that the Dawber data sets might be linear on time scales on the order of a second, but over longer time periods they might exhibit nonlinear, chaotic behaviour.

Taking the above differences between data sets into consideration, it might be possible that sea clutter *is* a chaotic process, but that chaotic behaviour was not identified in any of the wavetank and Dawber data sets. From an academic perspective it would be interesting to carry out yet more data analysis in order to either further substantiate, or to disprove the claim made that sea clutter is chaotic, in order to reach a conclusion on whether a chaotic or stochastic model is

more appropriate for sea clutter. However, in Chapter 2 it was pointed out that chaotic invariants cannot be used to distinguish between chaotic and stochastic processes, and without a method which can discriminate between them, the debate over whether or not sea clutter is chaotic becomes irrelevant. From a practical engineering perspective, however, the interest in carrying out further analysis on more sea clutter data sets (for example like those analysed by Haykin and his co-workers) would be to determine if the nonlinear signal processing techniques discussed in this thesis could be used in certain circumstances to improve radar target detection performance, regardless of whether clutter is actually chaotic or stochastic. Suggestions for such further practical analysis are given in section 9.3. It is suggested that the debate over whether clutter is chaotic or not should be avoided until such time as a classification methodology which can distinguish between chaotic and stochastic processes becomes available.

Filtering clutter before prediction analysis served to improve the performance of both a LP and a NLP. However, these results still reinforced the above finding that the wavetank and Dawber data sets have linear predictor functions. Noise was not responsible for the lack of evidence of chaotic behaviour.

Forward-backward prediction can be used to improve upon LP as well as NLP performance. However, a FBLP performed as well as, or better than a FB-RBFN.

A NLPD is not of any use for maritime surveillance radar, given that clutter has a linear predictor function: a LPD will perform as well as, or better, and would be preferred.

Even if chaotic clutter were encountered, the modelling of the dynamics of such a process appears to be irrelevant as such modelling can lead to a performance loss (in terms of a ROC) for a NLPD, relative to the performance of a NLPD which focused on minimising the MSE and not on the modelling of the underlying dynamics.

The modified Broomhead filtering method was not shown to be of any use on the (non-Gaussian) wavetank and clutter data sets. This technique may be of use for clutter data sets, or indeed other data in general, that have different non-Gaussian pdf's. This technique should not be ruled out for non-Gaussian processes in general. Furthermore, if chaotic clutter were encountered, then the positive results presented in section 8.4 would suggest that this technique could be used to improve maritime surveillance radar, by cancelling the clutter before initiating any detection signal processing routines.

9.3 Suggestions for further research

As no definitive conclusions on sea clutter modelling as a whole were proposed herein, and bearing in mind the contradictory nature of the results reported relative to those published by Haykin and his co-workers, it is suggested that further nonlinear signal processing analysis be carried out on clutter data collected at sea, for a wider set of radar and environmental parameters, than were used in the collection of the Dawber data sets. In particular, longer data records than 1.28 seconds should be analysed. The purpose of this study would not be to determine whether the collected clutter is chaotic, but rather, whether it would be worthwhile incorporating the nonlinear signal processing routines discussed in this thesis, into practical radar systems. For example, if it were found that in some circumstances a LPD could outperform a NLPD, whilst in others the reverse were true, a radar system could be designed to have the capability of switching between the two different algorithms.

Further experiments could also be carried out to investigate if pulse compression is responsible for the concealment of chaotic behaviour, or more generally, nonlinear behaviour in sea clutter. For example, it is suggested that nonlinear signal processing analysis be carried out on data collected from two different radars at the same site: one with a resolution achieved using pulse compression, and one with the same resolution which is achieved without the use of pulse compression. The resolution used should be of a similar order to that used by Haykin and his co-workers in their analysis of sea clutter.

Assuming a compound Gaussian stochastic model for sea clutter, further experiments could be performed to investigate the causes of the variation in speckle correlations with range, which were reported in the analysis of the wavetank data sets. This could perhaps lead to improved detection routines. For example, it might help to identify when to use a predictor-detector and when to use a less computationally involved CFAR detector or fixed threshold detector.

More analysis should be carried out on the possibility of using the modified Broomhead filter method to cancel chaotic noise from signals of interest. Such further analysis should include carrying out experiments using a number of different wideband chaotic (noise) time series, a number of different stochastic narrowband signals of interest (*i.e.* signals with different pdf's and bandwidths), and possibly even using deterministic narrowband signals of interest. Additionally, a study into the placement and number of notches, for the bandstop filter used in the modified Broomhead filter method, should be carried out. The application of the modified

Broomhead filter to maritime surveillance radar, should not be ruled out on the basis of the results presented herein. More clutter data sets should be examined, before reaching a conclusion on its suitability for maritime surveillance radar. In general, the modified Broomhead filter method should be tried on non-Gaussian noise processes encountered in different application areas, such as telecommunications: the technique may work on non-Gaussian processes with different pdf's to those of the wavetank and Dawber clutter data sets.

Appendix A
Sea clutter data

A.1 Introduction

The sea clutter data sets analysed for the work reported in this thesis are described in this appendix. The provider of all the data sets analysed was the Defence Evaluation and Research Agency (DERA). The general method used to collect these clutter data sets is described in section A.1. Section A.2 provides information on clutter data sets which were collected in a wavetank, these data sets will be referred to as the wavetank data sets. In section A.3, clutter data sets that were collected at sea, which will be referred to as the Dawber[1] data sets, are described.

A.2 Data collection method

The general method used to collect the clutter data sets, used for the work in this thesis, was as follows. A stationary land-based radar was operated in a dwelling mode, that is, with the antenna pointing towards a patch of the sea surface along a fixed direction. Figure A.1 depicts the resulting geometry of the data collection scenario.

The area of the clutter resolution cell A_{RC} is given by,

$$A_{RC} = \frac{c\tau_r}{2} R\theta \sec(\phi_r) \tag{A.1}$$

where R is the distance to the mid-range point of the resolution cell, θ is the beamwidth of the radar, ϕ_r is the grazing angle taken with respect to the centre of the radar beam, τ_r is the radar pulse width and c is the velocity of propagation.

[1] After the individual at DERA who provided the data sets, Bill Dawber.

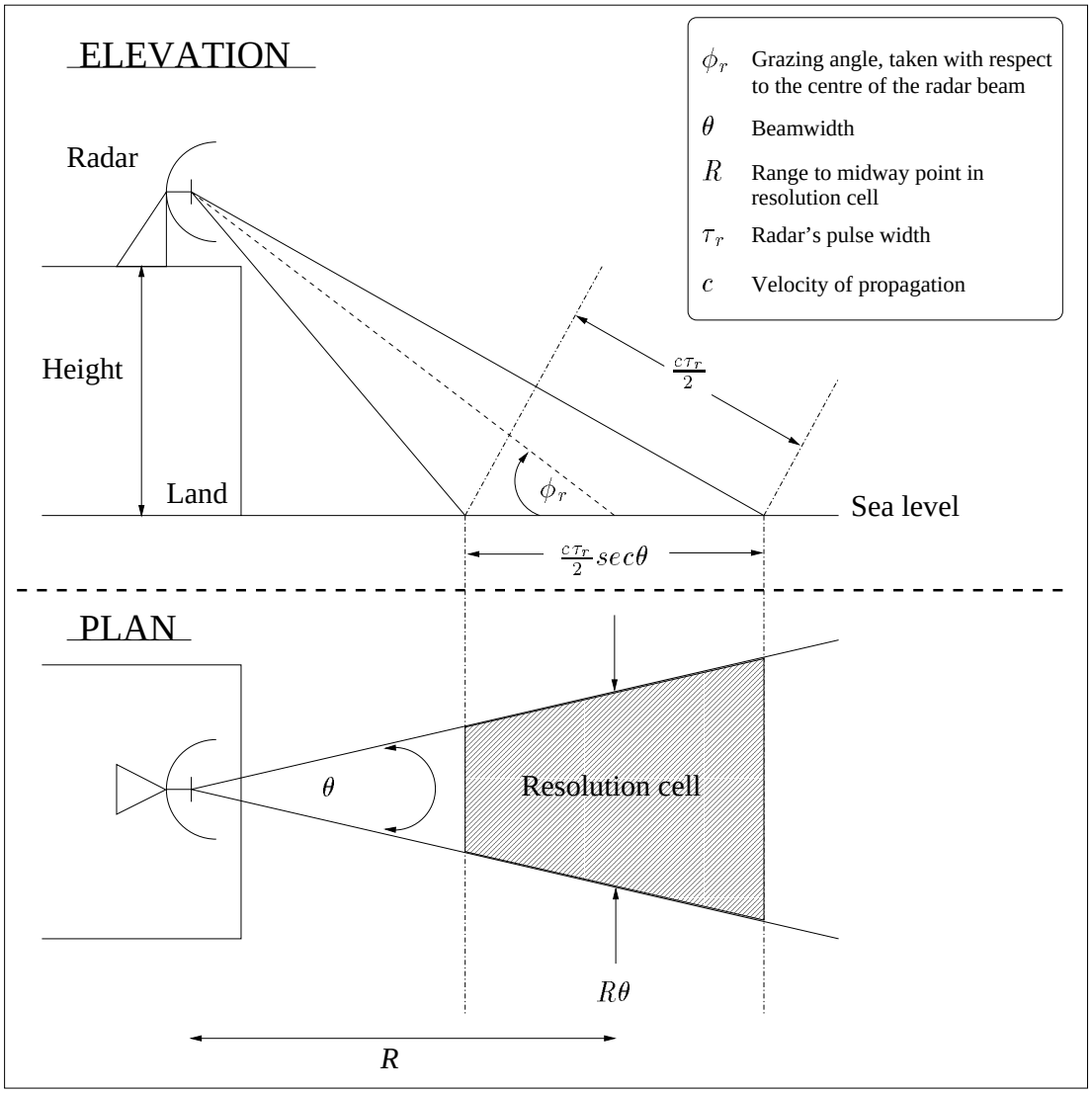


Figure A.1: *Collection of sea clutter data.*

As can be seen from Figure A.1 the dimensions of the resolution cell are determined by the radar pulse width τ_r in range, and by the beamwidth θ and range R across-range. Radar resolution is often only referred to in terms of range resolution τ_r.

A.3 The wavetank sea clutter data sets

The wavetank data sets were recorded in April 1998 as part of an experiment conducted by DERA Malvern and Racal Radar Defence Systems, at the large wavetank facility, in the ocean engineering laboratory of the University of California, Santa Barbara. The radar used was the Racal-Thorn mobile instrumented data acquisition system (MIDAS) [136].

A schematic of the wavetank experimental set-up is given in Figure A.2.

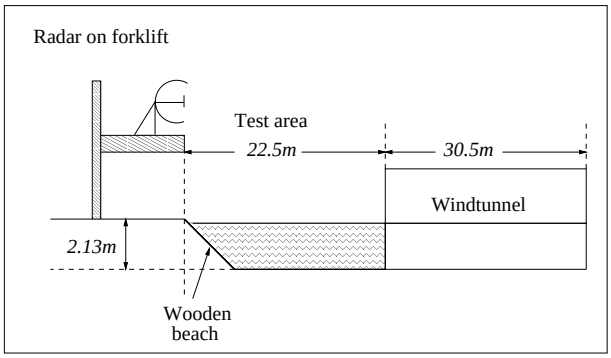


Figure A.2: *Schematic of wavetank experimental set-up.*

The wavetank is 53m long, 4.26m wide, and 2.13m deep. The wind tunnel extends 30.5m down the tank, leaving an open test section of 22.5m. A wooden beach at the test end of the tank reduces reflections. The wind tunnel can produce wind speeds of up to $12ms^{-1}$.

The MIDAS radar used pulse compression[2]. Pulse compression is a signal processing technique which allows a radar to use a long pulse to obtain a large radiated energy, but which also allows the range resolution of a short pulse to be achieved [7]. The range resolution of the radar was 0.3m (i.e. an effective pulse width of 2ns). Data was collected in 32 range cells, during wind speeds of $4ms^{-1}$ through to $12ms^{-1}$, in steps of $1ms^{-1}$. The data collected during a wind speed of Xms^{-1} will be referred to as the Xms^{-1} wavetank data set. Pulse to pulse transmit

[2]For more details on the specific implementation of pulse compression used by the MIDAS radar, see [136].

frequency agility was used, in a known (i.e. not randomised) sequence. The radar has a dual-polarised receiver. Only the transmit horizontal, receive horizontal (HH) data sets were made available for the work in this thesis. The effective pulse repetition frequency (PRF) of the radar was 1kHz. The grazing angle and beamwidth were 6^o and 5^o, respectively. There were 30,000 complex (i.e. coherent) samples collected in each range cell, for each wind speed data set.

A.4 The Dawber sea clutter data sets

These data sets were collected during experiments conducted by DERA Portsdownwest in January and February of 1994, 1995, and 1996, at Sennen Cove near Lands End, and also at Portsmouth (looking at the Isle of Wight) in December 1996. The radar used was the multi-band pulsed radar (MPR) designed and built by Roke Manor.

Two data sets from the experiments mentioned above were made available for the work reported in this thesis. Both of these data sets were collected without the use of pulse compression or polarisation agility. For both data sets the radar range resolution was $150m$ (i.e. a pulse width of $1\mu s$), and the PRF was 20kHz. The first data set, which will be called the Dawber-VV data set, was collected using vertical polarisation on transmit and receive during a wind speed of $12.8ms^{-1}$. The second data set, which will be called the Dawber-HH data set, was collected during a wind speed of $15.4ms^{-1}$. The grazing angle and beamwidth used in the collection of both data sets were 0.12^o and 6^o, respectively. There were 25,600 complex samples collected in each data set: these samples correspond to the temporal signal collected in one range cell, at a distance of 4km from the radar.

Appendix B
Sample moments

B.1 The method of moments

Given that x is a random variable from a random process with pdf $P_x(x)$, the k^{th} moment of such a random process is defined [137] as,

$$m_k = E[x^k] \tag{B.1}$$

where $E[.]$ is the expectation operator. If $x_1, x_2, \ldots, x_n$ are independently and identically distributed (i.i.d.) random variables, the k^{th} sample moment is defined [137] as,

$$\hat{m}_k = \frac{1}{n}\sum_{i=1}^{n} x_i^k \tag{B.2}$$

where $\hat{m}_k$ can be viewed as an estimate of the population moment m_k. The method of moments estimates parameters of distributions by finding expressions for them in terms of the lowest possible order moments and then substituting sample moments into the expressions [137].

B.2 Using the method of moments to estimate the parameters of the gamma distribution

A chi distributed process with pdf given by,

$$P_c(a_c|\nu, b) = \frac{2b^{2\nu}}{\Gamma(\nu)} a_c^{2\nu-1} \exp\left[-b^2 a_c^2\right]; \quad a_c \geq 0 \tag{B.3}$$

where $\Gamma(.)$ is the gamma function, b is the scale parameter and ν is the shape parameter, has k^{th} sample moments given [138] by,

$$\hat{m}_k = \frac{1}{b^k}\frac{\Gamma(\nu + (k/2))}{\Gamma(\nu)} \tag{B.4}$$

A gamma variate is simply the square of a chi variate. Therefore, a gamma distributed process with a pdf which is given by,

$$P_\Gamma(z|\nu, b) = \frac{b^{2\nu}}{\Gamma(\nu)} z^{\nu-1} \exp\left[-b^2 z\right]; \quad z \geq 0 \tag{B.5}$$

has k^{th} sample moments given by,

$$\hat{m}_k = \frac{1}{b^{2k}}\frac{\Gamma(\nu + k)}{\Gamma(\nu)} \tag{B.6}$$

Using equation B.6 and the fact that [98],

$$\Gamma(\nu + 1) = \nu\Gamma(\nu) \tag{B.7}$$

the first and second sample moments of the gamma distribution can be written as follows,

$$\hat{m}_1 = \frac{\nu}{b^2} \tag{B.8}$$

$$\hat{m}_2 = \frac{\nu(\nu + 1)}{b^4} \tag{B.9}$$

The above sample moments can be used to obtain the following shape and scale parameter estimates,

$$\hat{\nu} = \frac{\hat{m}_1^2}{\hat{m}_2 - \hat{m}_1^2} \tag{B.10}$$

$$\hat{b} = \frac{\hat{m}_1}{\hat{m}_2 - \hat{m}_1^2} \tag{B.11}$$

B.3 Using the method of moments to estimate the parameters of the K-distribution

A K-distributed process with pdf given by,

$$P_K(x|\nu, c) = \frac{4c}{\Gamma(\nu)}(cx)^{\nu} K_{\nu-1}(2cx); \quad x \geq 0 \tag{B.12}$$

where $\Gamma(.)$ is the gamma function, K_ν is a modified Bessel function of order ν, c is the scale parameter, and ν is the shape parameter, has k^{th} sample moments given [138] by,

$$\hat{m}_k = \frac{1}{c^k} \frac{\Gamma(\nu + (k/2))\Gamma(1 + (k/2))}{\Gamma(\nu)} \tag{B.13}$$

The second and fourth sample moments of the K-distribution can be used to obtain the following estimates for the shape and scale parameters,

$$\hat{\nu} = \left[\frac{4 - \frac{\hat{m}_4}{\hat{m}_2^2}}{\frac{\hat{m}_4}{\hat{m}_2^2} - 2} \right] + 1 \tag{B.14}$$

$$\hat{c} = \sqrt{\frac{\hat{\nu}}{\hat{m}_2}} \tag{B.15}$$

Appendix C
Tough and Ward simulation technique

C.1 Proofs involving the IIR correlation filter

C.1.1 Infinite series representation of the filter output

The IIR filter used to correlate Gaussian variates for the generation of correlated gamma variates, in Tough and Ward's simulation technique, is described by the following difference equation,

$$g'(n) = \rho g'(n-1) + \sqrt{1-\rho^2}\,g(n) \tag{C.1}$$

where $g'(n)$ is the correlated Gaussian output, $g(n)$ is the uncorrelated Gaussian input, and ρ is the correlation coefficient of the IIR filter.

The impulse response of this filter can be obtained by taking the z-transform of equation (C.1), and re-arranging to obtain the transfer function of the filter as follows,

$$G'(z) = \frac{\sqrt{1-\rho^2}}{1-\rho z^{-1}} G(z) \tag{C.2}$$

$$\Rightarrow H(z) = \frac{G'(z)}{G(z)} \tag{C.3}$$

$$\Rightarrow H(z) = \frac{\sqrt{1-\rho^2}\,z}{z-\rho} \tag{C.4}$$

from which the impulse response is obtained by taking the inverse z-transform as follows,

$$h(n) = \rho^n \sqrt{1-\rho^2} \tag{C.5}$$

The impulse response can be used to express the output of the IIR filter in terms of an infinite series, as shown below.

$$g'(n) = \sum_{k=0}^{\infty} h(k)g(n-k) \tag{C.6}$$

$$\Rightarrow g'(n) = \sqrt{1-\rho^2}\sum_{k=0}^{\infty} \rho^k g(n-k) \tag{C.7}$$

C.1.2 Mean proof

It can be simply shown that given a zero mean Gaussian input to the IIR filter, the correlated Gaussian output will also have a zero mean. Using equation (C.7), the mean of the IIR filter's output can be written as follows,

$$E[g'(n)] = E[\sqrt{1-\rho^2}\sum_{k=0}^{\infty} \rho^k g(n-k)] \tag{C.8}$$

$$= \sqrt{1-\rho^2}\sum_{k=0}^{\infty} \rho^k E[g(n-k)] \tag{C.9}$$

remembering that $0 \leq \rho < 1$, then using the result that the sum of a geometric series is convergent [139], it can be written that,

$$\sum_{k=0}^{\infty} \rho^k = \frac{1}{1-\rho} \tag{C.10}$$

and therefore,

$$E[g'(n)] = \frac{\sqrt{1-\rho^2}}{1-\rho} E[g(n-k)] \tag{C.11}$$

Assuming that $g(n)$ is wide-sense[1] stationary, then from equation (C.11), it can be seen that if $E[g(n)] = 0$, then $E[g'(n)] = 0$.

C.1.3 Variance proof

The following proof demonstrates that the variance of the output correlated Gaussian process of the IIR filter is the same as the variance of the input uncorrelated Gaussian process, given that the input Gaussian process has a mean value of zero. Using equation (C.7), the mean square value of the IIR filter's output can be written as follows,

$$E[(g'(n))^2] = E[(\sqrt{1-\rho^2}\sum_{k=0}^{\infty}\rho^k g(n-k))^2] \tag{C.12}$$

and since,

$$E[g(n)g(n+l)] = \begin{cases} 1 & \text{if } l = 0 \\ 0 & \text{otherwise} \end{cases}$$

then equation (C.12) can be re-written as,

$$E[(g'(n))^2] = 1-\rho^2\sum_{k=0}^{\infty}\rho^{2k}E[(g(n-k))^2] \tag{C.13}$$

Since,

$$\sum_{k=0}^{\infty}\rho^{2k} = \frac{1}{1-\rho^2} \tag{C.14}$$

then,

[1] *i.e.* the first and second order statistics of $g(n)$ do not change with time.

$$E[(g'(n))^2] \quad = \quad E[(g(n-k))^2] \tag{C.15}$$

Assuming that $\{g(n)\}$ is wide-sense stationary, then,

$$E[(g'(n))^2] \quad = \quad E[(g(n))^2] \tag{C.16}$$

and remembering that the variance of $\{g(n)\}$, σ_g^2, can be defined as,

$$\sigma_g^2 \quad = \quad E[(g(n))^2] - (E[g(n)])^2 \tag{C.17}$$

then it can be seen from equations (C.16) and (C.17) that when $\{g(n)\}$ has a zero mean value, the variance of $\{g(n)\}$ is equal to the variance of $\{g'(n)\}$.

C.1.4 ACF proof

Using equation (C.7), the ACF of the correlated Gaussian sequence at the output of the IIR filter can be written as follows,

$$\begin{aligned} R_{g'g'}(l) &= E[g'(n)g'(n+l)] \\ &= (1-\rho^2)\sum_{k_1=0}^{\infty}\rho^{k_1}\sum_{k_2=0}^{\infty}\rho^{k_2}E[g(n-k_1)g(n-k_2+l)] \end{aligned} \tag{C.18}$$

since,

$$E[g(n)g(n+l)] = \begin{cases} 1 & \text{if } l=0 \\ 0 & \text{otherwise} \end{cases}$$

then,

$$\begin{aligned} R_{g'g'}(l) &= (1-\rho^2)\sum_{k_1=0}^{\infty}\rho^{k_1}\rho^{k_1+l} \\ &= (1-\rho^2)\left(\sum_{k_1=0}^{\infty}\rho^{2k_1}\right)\rho^l \\ &= (1-\rho^2\left(\frac{1}{1-\rho^2}\right)\rho^l \\ &= \rho^l \end{aligned} \tag{C.19}$$

C.2 Changing the scale parameter of the gamma distribution

The following a simple proof demonstrates the ease with which the scale parameter of the gamma distribution can be changed.

The gamma pdf with scale parameter ν and scale parameter b equal to 1 is given by,

$$P_\Gamma(z|\nu, b=1) = \int_0^\infty \frac{1}{\Gamma(\nu)} z^{\nu-1}\exp\left[-z\right]dz; \quad z \geq 0 \tag{C.20}$$

Changing the variable of equation (C.20) from z to y where,

$$y = \frac{z}{b^2} \tag{C.21}$$

$$\Rightarrow dy = \frac{dz}{b^2} \tag{C.22}$$

and substituting equations (C.22) and (C.22) into equation (C.20) the following is obtained,

$$P_\Gamma(y = \frac{z}{b^2}) = \int_0^\infty \frac{1}{\Gamma(\nu)}(yb^2)^{\nu-1}\exp\left[-(yb^2)\right]\frac{dy}{b^2}$$

$$= \int_0^\infty \frac{b^{2\nu}}{\Gamma(\nu)}(y)^{\nu-1}\exp\left[-yb^2\right]dy \quad \text{(C.23)}$$

$$\text{(C.24)}$$

As can be seen by comparing equation (C.24) with equation (C.20), z is a gamma variate with a shape parameter ν and scale parameter 1, and y is a gamma variate with a shape parameter ν and scale parameter b. The above demonstrates how a gamma variate with one scale parameter can be scaled to obtain a gamma variate with a different scale parameter.

C.3 Maple script used to produce the discrete MNLT map

A simple *Maple* [97] script was used to solve equation (3.22) for $g'(n)$, and thus produce the discrete MNLT map. The adjustable parameters of the MNLT map are the minimum gamma variate value, the maximum gamma variate value, and the distance between successive gamma data points in the map. These parameters are controlled by *min*, *max*, and *step* respectively in the *Maple* script. A trial and error approach was adopted in the generation of an appropriate MNLT map. This was done by choosing values for *min*, *max* and *step*, then generating a MNLT map using these values in the *Maple* script. Then this map was used in a 3^{rd} order polynomial interpolation scheme, to produce gamma data, and any missed interpolations, due to *min* being too large, or *max* being too small were recorded. If there were any missed interpolations, then *min* and *max* were altered, a new MNLT map generated and new data simulated, and this new data was checked for any missed interpolations. This process was repeated until there were no missed interpolations. Once the map's upper and lower values had been selected, the parameter *step* was altered until the simulated gamma data had approximately the correct first 10 sample moments [66, 137].

The *Maple* script used to generate the MNLT map is now given on the following page.

```
> mnlt:=proc(filename,v,min,max,b,step);
> writeto(terminal):
> f:=(v,z,b) -> ((b ^ (2 * v))/GAMMA(v)) * Int( x ^ (v - 1) * exp(-1 * x * b * b), x=0..z):
> writeto(filename);
> i:=min;
> while i < max do printf('%.15g %.10g \n',evalf(f(v,i,b)), i);
> i:= i * step;
> od;
> close();
> end;
```

Appendix D
Householder transform

The Householder transformation [87, 116] is a reflection technique which can be used to solve the least squares (LS) problem. This transformation is numerically robust as it avoids estimating the inverse of the autocorrelation matrix directly.

The LS problem of interest for this thesis is the determination of the coefficients or weights of a linear predictor, or of a linear-in-the-parameter nonlinear predictor. The output $\hat{y}(n)$ of an N-tap linear predictor may be written as,

$$\hat{y}(n) = \sum_{i=0}^{N-1} h(n)x(n-i) \tag{D.1}$$

$$= \mathbf{h}^T\mathbf{x}(n) \tag{D.2}$$

where $\mathbf{h}$ is a N-long column vector of the impulse response sequence of the linear predictor, and $\mathbf{x}(n)$ is a column vector containing the last N elements of the input sequence. The LS estimate of the impulse response vector is that which minimises the sum of squared error cost function $\xi(k)$ evaluated over $k+1$ data points where,

$$\xi(k) = \sum_{n=0}^{k} e^2(n) \tag{D.3}$$

and the error is given by,

$$e(n) = y(n) - \hat{y}(n) \tag{D.4}$$

The standard solution to this minimisation problem is based on forming the auto-correlation

matrix $\mathbf{R}_{xx}$, and a cross-correlation vector $\mathbf{r}_{xy}(k)$. Thus,

$$\mathbf{R}_{xx}(k)\,\mathbf{h}(k) = \mathbf{r}_{xy}(k) \tag{D.5}$$

where,

$$\mathbf{R}_{xx}(k) = \sum_{n=0}^{k} \mathbf{x}(n)\mathbf{x}^T(n) \tag{D.6}$$

and,

$$\mathbf{r}_{xy}(k) = \sum_{n=0}^{k} \mathbf{x}(n)y(n) \tag{D.7}$$

To avoid having to invert the auto-correlation matrix in order solve the above LS problem, the following approach may be used. First, consider a matrix $\mathbf{Q}$ which is orthonormal. Such a matrix has two important properties: it has an inverse which is equal to it's transpose,

$$\mathbf{Q}^{-1} = \mathbf{Q}^T \tag{D.8}$$

and also, the matrix $\mathbf{Q}$ is said to be able to reflect a vector, without changing the Euclidean length of that vector, i.e.,

$$|\mathbf{e}| = |\mathbf{Q}\mathbf{e}| \tag{D.9}$$

Therefore, the cost function of equation (D.3) can be re-written as,

$$\xi(k) = |\mathbf{Q}(k)\mathbf{e}(k)| \tag{D.10}$$

Using equations (D.4), (D.2) and (D.10), the following can be written,

$$\mathbf{Q}(k)\mathbf{e}(k) = \mathbf{Q}(k)\mathbf{y}(k) - \mathbf{Q}(k)\mathbf{X}(k)\mathbf{h}(k) \tag{D.11}$$

where,

$$\mathbf{X}^T(k) = [\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(k)] \tag{D.12}$$

and $\mathbf{X}(k)$ is referred to as the data matrix [116].

The orthonormal transformation of the data matrix yields an upper triangular matrix,

$$\mathbf{Q}(k)\mathbf{X}(k) = \begin{bmatrix} \mathbf{U}(k) \\ \mathbf{0} \end{bmatrix} \tag{D.13}$$

where $\mathbf{U}(k)$ is an $N \times N$ upper triangular matrix. Introducing two N-element vectors $\mathbf{u}(k)$ and $\mathbf{v}(k)$ yields,

$$\mathbf{Q}(k)\mathbf{y}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{v}(k) \end{bmatrix} \tag{D.14}$$

The error vector can now be expressed as,

$$\mathbf{Q}(k)\mathbf{e}(k) = \begin{bmatrix} \mathbf{u}(k) - \mathbf{U}(k)\mathbf{h}(k) \\ \mathbf{v}(k) \end{bmatrix} \tag{D.15}$$

The Euclidean length of the error vector will be minimised when,

$$\mathbf{u}(k) = \mathbf{U}(k)\mathbf{h}(k) \tag{D.16}$$

which determines the impulse response vector $\mathbf{h}(k)$. The elements of the impulse response vector can be calculated by a simple back substitution operation because $\mathbf{U}(k)$ is an upper triangular matrix. The minimum error is directly available as,

$$|\mathbf{e}(k)| = |\mathbf{v}(k)| \tag{D.17}$$

The transformation of a data matrix into an upper triangular matrix may be achieved by using Householder transformations. To demonstrate how this is achieved imagine that the triangularisation process has already been started, and that $\mathbf{R}$ is a rectangular data matrix whose first few columns are already in upper triangular form. Thus $\mathbf{R}$ may be partitioned as follows,

$$\mathbf{R} = \begin{bmatrix} \mathbf{U} & * \\ \mathbf{0} & \mathbf{R}_s \end{bmatrix} \tag{D.18}$$

where $\mathbf{U}$ is a square upper triangular matrix, $\mathbf{R}_s$ is a rectangular matrix, and $*$ contains terms which are not relevant. The next step is to find an orthonormal matrix which preserves the existing upper triangular structure and forces one extra column of $\mathbf{R}$ to be upper triangular. The following matrix may be used,

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_s \end{bmatrix} \begin{bmatrix} \mathbf{U} & * \\ \mathbf{0} & \mathbf{R}_s \end{bmatrix} = \begin{bmatrix} \mathbf{U} & * \\ \mathbf{0} & \mathbf{Q}_s\mathbf{R}_s \end{bmatrix} \tag{D.19}$$

This operation leaves the first few rows and columns of $\mathbf{R}$ unaffected, and reduces the problem to one of finding a unitary matrix $\mathbf{Q}_s$, i.e.,

$$\mathbf{Q}_s^{-1} = \mathbf{Q}_s^H \tag{D.20}$$

which when applied to $\mathbf{R}_s$ will convert it's first column $\mathbf{r}_1$ to all zero elements, except for the first entry, i.e. the first row, first column of $\mathbf{R}_s$, which will be denoted as $\mathbf{R}_s(1,1)$. The Householder method for doing this is summarised below:

1. Define a vector $\mathbf{u}$,

   $\mathbf{u} = \mathbf{r}_1 + \text{sign}(\mathbf{R}_s(1,1))|\mathbf{r}_1|\mathbf{1}$

   where $\mathbf{1} = [1, 0, 0, \ldots 0]^T$. This operation only changes one element of $\mathbf{r}_1$ to form $\mathbf{u}$. The sign function is defined as sign$(x) = 1$ if $x \geq 0$ and sign$(x) = -1$ otherwise.

2. Use $\mathbf{u}$ to define a vector $\hat{\mathbf{u}}$ which is used to implement the transformation,

   $\hat{\mathbf{u}} = \frac{2\mathbf{u}}{|\mathbf{u}|^2}$

3. Apply the transformation to $\mathbf{r}_1$,

   $\mathbf{Q}_s\mathbf{r}_1 = \mathbf{r}_1 - \mathbf{u} = -\text{sign}(\mathbf{R}_s(1,1))|\mathbf{r}_1|\mathbf{1}$

   The first element of $\mathbf{Q}_s\mathbf{r}_1$ is -sign$(\mathbf{R}_s(1,1))\sigma$, where $\sigma = \sqrt{\sum_i (\mathbf{R}_s(i,1))^2}$. The rest of the elements are zero by definition.

4. Apply the transformation to the remaining columns $\mathbf{r}_j$ of $\mathbf{R}_s$,

   $\mathbf{Q}_s\mathbf{r}_j = \mathbf{r}_j - (\hat{\mathbf{u}}^T\mathbf{r}_j)\mathbf{u}$

Although the Householder transformation is defined by a matrix $\mathbf{Q}_s$, the matrix is not stored explicitly and the computational requirements of steps 1 to 4 are much less than a matrix multiplication. Note also that only one square root operation (step 3) is required for each column of the matrix $\mathbf{R}$.

For the training of the linear and nonlinear predictors used for this thesis a block approach was adopted. This strategy is briefly explained now. If a predictor has N coefficients then the data matrix $\mathbf{X}(k)$ will have N columns. The number of rows will depend on the length of the data record used to train the filter. The larger the data record, the better the quality the LS estimate of the coefficients of the predictor will be. However, a large data record will require a large data matrix, and hence may impose excessive memory requirements on the processor. The following strategy was adopted as a reasonable compromise between memory requirement and performance for training and up-dating the coefficients of the predictor.

To start the process a $2N \times N$ element matrix is constructed from $2N$ consecutive input vectors $\mathbf{x}(n)$, i.e. from $n = 0$ to $n = N - 1$. A corresponding $2N \times 1$ target vector y is also assembled. This defines the LS problem:

$$\mathbf{X}\mathbf{h} = \mathbf{y}$$

The above equation defines the major storage requirements for the algorithm. A series of Householder transformations defined by the $2N \times 2N$ unitary matrix $\mathbf{Q}$ are applied to both sides of the equation:

$$\mathbf{Q}\mathbf{X}\mathbf{h} = \mathbf{Q}\mathbf{y}$$

and so,

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix} \mathbf{h} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$$

The $N \times N$ upper triangular matrix $\mathbf{U}$ and the $N \times 1$ vector $\mathbf{u}$ form a set of simultaneous equations which can be solved by back substitution to form the estimate $\mathbf{h}$. This estimate can then be updated using blocks of N consecutive vectors $\mathbf{x}(n)$ and a corresponding block of N target samples $y(n)$. For example, the first block to use as an update would range from $n = 2N$ to $n = 3N - 1$. The N vectors are placed in an $N \times N$ matrix $\mathbf{X}_+$ i.e.,

$$\mathbf{X}_+ = \begin{bmatrix} \mathbf{x}^T(2N) \\ \mathbf{x}^T(2N+1) \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}^T(3N-1) \end{bmatrix}$$

In a similar manner the N new target samples are collected into an $N \times 1$ vector $\mathbf{y}_+$,

$$\mathbf{y}_+ = \begin{bmatrix} y(2N) \\ y(2N+1) \\ . \\ . \\ . \\ y(3N-1) \end{bmatrix}$$

The matrix $\mathbf{X}_+$ and vector $\mathbf{y}_+$ are used to overwrite the lower half of the triangularised LS equations from the previous stage. Thus,

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{X}_+ \end{bmatrix} \mathbf{h} = \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_+ \end{bmatrix}$$

Householder transformations are then applied to both sides of this equation to form a new upper triangular set,

$$\mathbf{Q}_+ \begin{bmatrix} \mathbf{U} \\ \mathbf{X}_+ \end{bmatrix} \mathbf{h} = \mathbf{Q}_+ \begin{bmatrix} \mathbf{u} \\ \mathbf{y}_+ \end{bmatrix}$$

and so,

$$\begin{bmatrix} \mathbf{U}_+ \\ \mathbf{0} \end{bmatrix} \mathbf{h} = \begin{bmatrix} \mathbf{u}_+ \\ \mathbf{v}_+ \end{bmatrix}$$

Back substitution produces the updated predictor coefficients. This procedure can be repeated as each new block of N input vector and target samples becomes available.

Appendix E
Wavetank clutter data set correlation dimension estimates

Windspeed [ms^{-1}]

Range gate	4	5	6	7	8	9	10	11	12
0	3.694730	3.670330	3.673680	4.411110	3.964700	4.351640	4.224720	5.415380	4.750000
1	3.634410	3.797870	3.630430	4.325580	4.139530	4.300000	4.130430	3.676920	5.101260
2	3.683670	3.691490	3.619560	4.417580	4.895350	4.761900	4.833330	2.819440	5.256750
3	3.610520	3.691490	3.634410	4.272720	4.938270	5.679490	4.845230	2.985910	5.647880
4	3.688170	3.670330	3.584270	4.280900	4.682350	5.671420	5.272720	3.771430	6.104470
5	3.673910	3.630430	3.692310	3.595500	4.602560	5.342850	5.846150	4.338230	5.794110
6	3.683670	3.641300	3.644440	4.333330	4.764700	6.069440	5.784610	4.757570	5.179100
7	3.663160	3.549450	3.600000	4.393260	4.939760	5.641790	5.553840	3.869560	4.791040
8	3.706520	3.593750	3.600000	4.409090	4.226190	6.173910	4.671640	3.514280	4.298500
9	3.625000	3.619560	3.560440	4.229880	4.287350	5.774640	5.031250	4.676920	3.642860
10	3.634410	3.648930	3.547370	4.268810	4.117020	5.216210	4.696970	4.397060	3.128570
11	3.634410	3.663160	3.642100	3.652170	5.011900	4.821430	5.015870	5.758060	3.702700
12	3.692310	3.670330	3.659340	3.666660	4.372090	5.035710	5.388060	6.393940	4.454540
13	3.747250	3.695650	3.626370	3.659340	4.459770	5.068190	5.628570	6.507460	4.784610
14	3.630430	3.648930	3.711110	3.695650	4.223530	4.160920	4.955220	6.436630	4.194440
15	3.606380	3.663040	3.617020	3.587630	4.244440	4.204540	4.527780	6.189200	4.731340
16	3.709670	3.694730	3.681320	3.626370	4.423910	4.313950	6.046870	6.041100	5.384610
17	3.645160	3.715790	3.702130	3.684210	4.483140	4.455550	6.360010	5.837840	4.985290
18	3.595740	3.673910	3.680850	3.634410	3.608690	3.637360	5.729730	5.630130	5.000000
19	3.720430	3.612900	3.694730	3.614580	3.597820	4.406590	6.236850	6.259750	5.906250
20	3.677780	3.648930	3.666660	3.589470	3.631580	4.385540	6.133340	5.972220	6.441170
21	3.741930	3.698920	3.726310	3.709670	3.702130	4.561790	5.958900	5.987820	6.246580
22	3.723400	3.666660	3.677080	3.677780	3.706520	4.565220	6.148170	5.475620	6.797320
23	3.652630	3.617020	3.709670	3.638300	3.705260	4.400000	5.809540	4.865170	7.102600
24	3.750000	3.736840	3.724490	3.659340	3.663040	4.516850	5.011900	4.922230	6.109590
25	3.666660	3.608690	3.692310	3.602150	3.642100	4.488890	5.255570	4.965520	6.320530
26	3.666660	3.736840	3.704080	3.778940	3.742270	3.663160	5.174420	5.244460	5.313260
27	3.691490	3.712760	3.744680	3.702130	3.789470	4.611110	4.395350	4.306820	5.512200
28	3.688890	3.722220	3.694730	3.736260	4.543480	4.611760	5.246910	5.322240	5.860490
29	3.621050	3.744680	4.588890	3.741930	4.483140	4.528090	5.195410	5.284100	5.271600
30	3.692310	3.731960	4.489360	4.494380	5.127910	4.384610	5.170460	5.035290	5.211770
31	3.600000	3.714280	4.433330	4.427090	5.536590	5.617290	5.065940	5.937510	5.185590

Table E.1: *Maximum likelihood correlation dimension estimates, D_{ML}, for the wavetank clutter data sets.*

Appendix F
CD

Included with this thesis is a CD, the contents of which are now described. The initial linear and nonlinear prediction results for the $12ms^{-1}$ wavetank range gate data sets, which were discussed in section 5.2.2, have been recorded onto the CD in the form of a postscript document. Linear prediction results are given for 5, 10, 20 and 40 tap LP's. Results are presented for cubic VSFP's, with embedding dimensions of 5 and 10, and an embedding delay of 1 sample for each case of embedding dimension. NRBFNP-RSC results are shown for embedding dimensions of 5, 10, 20 and 40. Additionally, results are shown for NRBFNP-OAKM's, UNRBFNP-RSC's, and UNRBFNP-OAKM's with an embedding dimension of 20. For each RBFNP result, an embedding delay of 1 sample, and 100 Gaussian kernels were used.

Appendix G
Original publications

Some of the work in this thesis has been reported in the following publications, those marked by † are reprinted in this appendix:

- † M. Cowper and B. Mulgrew, "Performance of radial basis function networks as predictors for sea clutter", in IEE Radar and Sonar Signal Processing, Peebles, Scotland, July 1998.
- † M. Cowper and B. Mulgrew, "Nonlinear processing of high resolution radar sea clutter", in IEEE International Joint Conference on Neural Networks, Washington D.C., USA, July 10-16, 1999.
- C.P. Unsworth, M.R. Cowper, S. McLaughlin, and B. Mulgrew, "Detection of nonlinearity in a time series: by synthesis of surrogate data using a Kolmogorov-Smirnoff tested, hidden Markov model", 33rd Asilomar Conference on Signals, Systems and Computers, October 1999, CA, USA.
- M. R. Cowper and B. Mulgrew, "The application of a nonlinear inverse noise cancellation technique to maritime surveillance radar", accepted for publication in the 10^{th} IEEE Signal Processing Workshop on Statistical Signal and Array Processing, Pocono Manor, PA, USA, August 14-16, 2000.
- M. R. Cowper and B. Mulgrew, "Nonlinear processing of chaotic signals", accepted for publication in Euroattractor 2000, Warsaw, Poland, June 6-15, 2000.
- C.P. Unsworth, M. R. Cowper, S. McLaughlin, and B. Mulgrew, "Chaotic invariants detect chaos in the stochastic compound K-distribution model of radar sea clutter", accepted for publication in the 10^{th} IEEE Signal Processing Workshop on Statistical Signal and Array Processing, Pocono Manor, PA, USA, August 14-16, 2000.
- B. Mulgrew, C. P. Unsworth, M. R. Cowper, and S. McLaughlin, "Sea clutter and chaos: improved surrogate-data tests", accepted for publication in EUSIPCO 2000, Tampere, Finland, 5-8 September, 2000.

PERFORMANCE OF RADIAL BASIS FUNCTION NETWORKS AS PREDICTORS FOR SEA CLUTTER

M R Cowper, B Mulgrew

University of Edinburgh, U.K.

INTRODUCTION

The performance of radars used in marine environments is limited by sea clutter. The suppression of this clutter is necessary in order to improve the target detection capability of such radars.

Currently, there are two conflicting views on how to model sea clutter. The traditional approach has been to view sea clutter as a stochastic process. Evidence has suggested that the stochastic process which best models sea clutter is the (non-Gaussian) compound K-distribution, Ward [1].

However, recently an argument has been put forward which suggests that sea clutter is not in fact a stochastic process, but rather, it is a chaotic process, Haykin et al. [2].

These non-Gaussian descriptions provide the motivation to investigate the possibility of using non-linear techniques to suppress sea clutter, rather than the conventional linear techniques used to suppress Gaussian processes.

The objective of this paper is to investigate the use of fast training methods for radial basis function (RBF) predictor networks, Haykin [3]. The fast training methods will enable the RBF predictor networks to operate in real time or quasi-real time. The ability of such networks to predict sea clutter, as compared with linear techniques, will be reported. This comparison will be used to consider the effectiveness of using such non-linear predictors to suppress sea clutter.

RBF PREDICTION

Prediction

In its simplest form, a prediction problem is based on a time series $\{x(n)\}$. Given an embedding vector $\mathbf{x}(n)$, which contains N consecutive samples of the time series, i.e. :

$$\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$$

This is used to form an estimate $\hat{x}(n+1)$ of the next data sample, $x(n+1)$, by constructing a predictor function $f()$, where

$$\hat{x}(n+1) = f(\mathbf{x}(n))$$

RBF predictor

The RBF predictor is used to approximate the predictor function $f(\mathbf{x}(n))$, discussed above. The structure of the RBF predictor is shown in figure 1. It consists of N source nodes, M centres (or hidden units), and M weights.

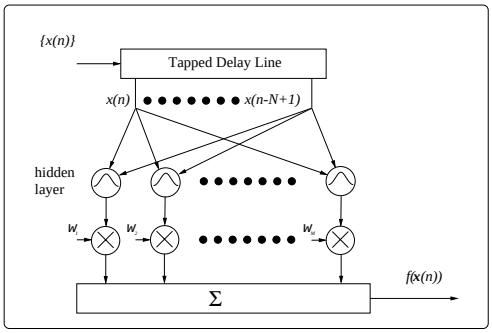


Figure 1: *RBF predictor.*

The RBF predictor's approximation of the predictor function, for the input vector $\mathbf{x}(n)$, is given by the following equation:

$$\hat{f}(\mathbf{x}(n)) = \sum_{i=1}^{M} w_i \phi(\|\mathbf{x}(n) - \mathbf{c}_i\|) \qquad (1)$$

where M is the number of centres, $\phi()$ is some non-linear function, $\|.\|$ is a distance measure, $\mathbf{c}_i$ is the position of the i^{th} centre in N-dimensional space, and w_i is the weight at the output of the i^{th} centre.

The nonlinear function chosen for this work was the Gaussian function. The distance measure used was the Euclidean distance.

FAST TRAINING

Training of the RBF networks involves selecting centres and training the output layer of weights.

To achieve centre selection in real time or quasi-real time, two options are available. Firstly, the centres can be chosen at random from the training data available to the RBF. Secondly, the k-means algorithm, can be used. The optimal adaptive k-means algorithm was chosen for this study, Chinrungrueng [4].

Training The RBF's Output Layer Weights

The obvious choices for training the output layer weights are the least mean square (LMS) algorithm, or a least squares (LS) approach. However, the LMS convergence time is too slow to allow for real or quasi-real time implementations. The LS approach adopted in this paper is the Householder transform, Haykin [5].

The Householder transformation is a reflection technique which can be used to solve the LS problem. The algorithm in this paper is used in an adaptive block least squares approach. If an additional block of new data is available, the LS solution will be updated in a computationally efficient manner. The algorithm is numerically robust, because it avoids estimating the inverse of the autocorrelation matrix directly, Steinhardt [6].

PREDICTION OF A STANDARD CHAOTIC SIGNAL

The fast training techniques are initially examined using the RBF to predict a standard chaotic signal. The signal chosen was generated using the Lorenz set of equations, Lorenz [7].

The two different centre selection methods described above are compared using the mean square error (MSE) performance measure. The effect of normalising the outputs of the RBF centres is also observed. In addition, the effects on MSE performance measure, of varying the following parameters are reported: embedding dimension, embedding delay, and number of centres.

PREDICTION OF SEA CLUTTER DATA

Finally, the performance of the RBF predictor, using the fast training techniques, is reported for sea clutter data. The feasibility of using such a network to cancel clutter is considered.

ACKNOWLEDGEMENTS

The authors are grateful to the Defence Research Evaluation Agency (DERA), for supplying the sea clutter data used in this study. M R Cowper would like to thank GEC Marconi Avionics for their financial support.

References

[1] K. D. Ward, "Compound representation of high resolution sea clutter," *Electron. lett.*, vol. 17, no. 16, pp. 561–563, 1981.

[2] S. Haykin and S. Puthusserypady, "Chaotic dynamics of sea clutter," *Chaos*, vol. 7, no. 4, pp. 777–802, 1997.

[3] S. Haykin, *Neural Networks*. Macmillan, 1994.

k-means algorithm with dynamic adjustment of learning rate," *IEEE Transactions on Neural Networks*, vol. 6, pp. 157–169, January 1995.

[5] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2nd ed., 1991.

[6] A. O. Steinhardt, "Householder transforms in signal processing," *IEEE ASSP Magazine*, pp. 4–12, 1988.

[7] E. N. Lorenz, "Atmospheric predictibility as revealed by naturally occurring analogues," *J. Atmos. Sci.*, 1969.

Nonlinear processing of high resolution radar sea clutter

Mark R. Cowper and Bernard Mulgrew

Signals and Systems Group, Department of Electronics and Electrical Engineering, University of Edinburgh
The King's Buildings, Edinburgh, EH9 3JL, Scotland, UK
mrc@ee.ed.ac.uk, bernie@ee.ed.ac.uk

Abstract

This work is concerned with investigating whether or not nonlinear predictor networks can be used to improve the performance of high resolution surveillance radars which are used to detect targets on, or near, the sea surface. Prediction and detection results are presented for new sea clutter data sets.

1 Introduction

The performance of surveillance radars used in marine environments is limited by sea clutter, the unavoidable radar returns from the sea surface. Low resolution clutter returns are widely accepted to have Gaussian statistics. However, the statistics of high resolution sea clutter deviate from the Gaussian case. These non-Gaussian clutter returns are often characterised by frequently occurring large signal values (or spikes), which can be mistaken for target signals.

The traditional approach to clutter modelling and detector design has been to use a stochastic process to model sea clutter. Evidence has suggested that the process which best models sea clutter is the compound K-distribution [1]. However, research has been carried out which suggests that sea clutter is not in fact a stochastic process, but rather, it is a chaotic process [2–4]. It should be noted that the categorisation of sea clutter as a chaotic process, using the techniques described in [3, 4], has been questioned by other researchers [5, 6].

A nonlinear predictor-detector has been shown to perform better than a standard detector in [3], although no linear predictor-detector comparison was carried out in that study. The purpose of this paper is to investigate whether nonlinear predictor networks can be used to improve high resolution radar detection of targets on, or near, the sea surface. This will be done using the sea clutter data sets described in *section 2*. If sea clutter is chaotic, then nonlinear predictors will be able to exploit this property. If it is not then nonlinear predictors may be able to exploit the widely accepted non-Gaussian nature of high resolution sea clutter.

This work was supported by Marconi Avionics, DERA, and EPSRC.

2 Sea clutter data

Sea clutter data sets have been collected using stationary, land-based radars that operate in a dwelling mode, that is, with the antenna pointing towards a patch of the sea surface along a fixed direction.

2.1 Wavetank data

Several sea clutter data sets have been collected for this analysis, in a wavetank. The wavetank allows data to be collected during a range of controlled wind speeds. The length of the wavetank is $50m$: the first $20m$ were used to develop the waves, the remaining $30m$ were used as the data collection area. The wind direction, used for all data sets collected, was towards the radar. The data was collected in 32 range cells. Pulse compression was used to achieve a range resolution of $0.3m$. Data was collected during wind speeds of $4ms^{-1}$ through to $12ms^{-1}$ in steps of $1ms^{-1}$. Pulse to pulse transmit polarisation agility was used. The radar had a dual-polarisation receive capability. Only the transmit horizontal, receive horizontal (HH) data sets have been analysed in this paper. The effective pulse repetition frequency (PRF) of the radar was $1kHz$. The grazing angle and beamwidth were 6^o and 5^o, respectively. There were $30,000$ complex samples collected in each range cell, for each wind speed data set. Results for the non-coherent[1] samples are presented in this paper.

2.2 Dawber data

The other two data sets analysed in this paper were collected at sea. Both of these data sets were collected using a radar that did not employ pulse compression or polarisation agility. For both data sets the radar range resolution was $150m$, and the PRF was $20kHz$. The first data set, Dawber-VV, was collected using vertical polarisation on transmit and receive, during a wind speed of $12.8ms^{-1}$. The second data

[1] A non-coherent sample is the modulus of a complex sample.

set, Dawber-HH, was collected using horizontal polarisation on transmit and receive, during a wind speed of $15.4ms^{-1}$. The grazing angle and beamwidth used in each case were 0.12^o and 6^o, respectively. There were $25,600$ complex samples collected in each data set: these samples correspond to the temporal signal collected in one range cell, at a distance of $4km$ from the radar. Results for the non-coherent samples are presented in this paper.

3 Prediction

The radial basis function (RBF) network and the Volterra series network have been chosen to implement the nonlinear predictors. These networks are briefly discussed below.

3.1 Prediction problem

A prediction problem is based on a time series $\{x(n)\}$. Given a vector $\mathbf{x}(n)$ from this time series, with an embedding dimension N and an embedding delay τ, i.e.

$$\mathbf{x}(n) = [x(n), x(n-\tau), ..., x(n-(N-1)\tau)]^T$$

an estimate $\hat{x}(n+1)$ of the next data sample, $x(n+1)$, is formed by constructing a predictor function $f()$, where

$$\hat{x}(n+1) = f(\mathbf{x}(n)) \tag{1}$$

3.2 RBF predictor

An RBF network can be used to find the predictor function discussed above. The structure of an RBF network is shown in Figure 1. It consists of N source nodes, M centres (or hidden units), and M weights.

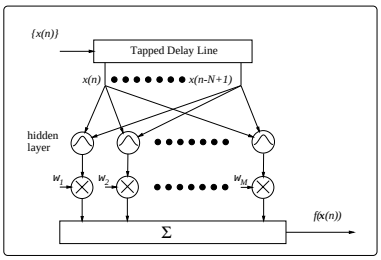


Figure 1: RBF network.

The RBF predictor function is given by,

$$f(\mathbf{x}(n)) = \sum_{i=1}^{M} w_i \phi(\|\mathbf{x}(n) - \mathbf{c}_i\|) \tag{2}$$

where M is the number of centres, $\phi()$ is some non-linear function (or kernel), $\|.\|$ is a distance measure, $\mathbf{c}_i$ is the position of the i^{th} centre in N-dimensional space, and w_i is the weight at the output of the i^{th} centre.

The nonlinear kernel chosen for this work was the Gaussian function, which is symbolised in Figure 1 by the bell-shaped curves in the hidden layer. Normalised [7] as well as unnormalised Gaussian kernels have been used. The distance measure used was the Euclidean distance. The positions of the centres were selected at random from the training data.

3.3 RBF embedding dimension and embedding delay

In order to reconstruct the dynamics of the clutter data, (assuming that the clutter data is the result of a nonlinear dynamical process), it is necessary to choose an appropriate number of RBF source nodes, N, usually termed the embedding dimension, and also an appropriate embedding delay.

3.3.1 Correlation dimension

An embedding dimension, N, should be chosen [8] such that:

$$N \geq 2d + 1 \tag{3}$$

where d is estimated using the correlation dimension [9]. If the correlation dimension estimate of d is not an integer value, it should be rounded up to the next integer value. For the analysis in this paper, the maximum likelihood correlation dimension, D_{ML}, [9] was used to estimate d.

3.3.2 Average mutual information

To determine the embedding delay of the clutter data, the average mutual information [10] was used. It has been suggested [11] that the optimum embedding delay is at the first minimum of the average mutual information, $I(T)$. The average mutual information between $x(n)$ and $x(n+T)$ is defined as:

$$I(T) = \sum_{n=1}^{K} P(x(n), x(n+T)) \times log_2 \left[\frac{P(x(n), x(n+T))}{P(x(n))P(x(n+T))} \right] \tag{4}$$

where $P(x(n), x(n+T))$ is the joint probability density for measurements $x(n)$ and $x(n+T)$.

3.4 Volterra series predictor

The truncated Volterra series (5) can also be used to find the predictor function discussed above.

$$f(\mathbf{x}(n)) = \sum_{m_1=0}^{N-1} h_1(m_1)x(n-m_1) \quad (5)$$
$$+ \sum_{m_1=0}^{N-1}\sum_{m_2=m_1}^{N-1} h_2(m_1,m_2)x(n-m_1)x(n-m_2) + ...$$
$$+ \sum_{m_1=0}^{N-1}\cdots\sum_{m_L=m_1}^{N-1} h_2(m_1,m_2,...,m_L)$$
$$\times x(n-m_1)...x(n-m_L)$$

The truncated Volterra series expansion is given above, for a zero mean process. N is the embedding dimension, L is the order of the Volterra series, and $h(.)$ is known as a Volterra kernel.

3.5 Prediction performance assessment

The prediction performance measure that has been used for this study is the normalised mean square error (*NMSE*), which is defined as:

$$NMSE = 10\log_{10}\left(\frac{1}{\sigma_y^2 J}\sum_{k=1}^{J}(y(k)-\hat{y}(k))^2\right) \quad (6)$$

where $y(k)$ is the desired predictor output, $\hat{y}(k)$ is the predictor's estimate of $y(k)$, and σ_y^2 is the variance of y over J. Prediction results were obtained by using training, testing and validation data sets. If the training data length was selected to be J samples, the next J samples immediately after the training data were selected as the testing data set. The next J samples immediately after the testing data set were selected as the validation data set.

4 Prediction of the wavetank data

The temporal signals in each range cell, for each wind speed data set, were analysed using a linear predictor and the nonlinear predictors described in *section 3*.

4.1 Prediction of the temporal signal in range cell 14 of the $12\mathrm{ms}^{-1}$ data set

The criteria discussed in *section 3.3* were used as a starting point to choose the RBF embedding dimension and embedding delay. For range cell 14 of the $12ms^{-1}$ data set, $D_{ML} = 4.19$. An embedding dimension of 12 was chosen. A plot of the mutual information for range cell 14 is given in Figure 2. As can be seen there are no obvious minima, the mutual information curve is monotonically decreasing.

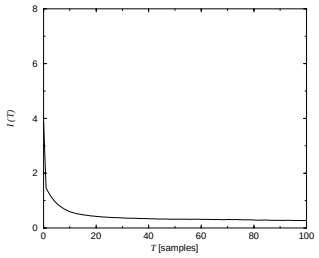


Figure 2: Mutual Information for range cell 14 of $12ms^{-1}$ data set.

Thus the criteria for selecting an embedding delay in *section 3.3.2* is of no use here, therefore an embedding delay of 1 sample was selected as a starting point. Results for a linear predictor, a cubic Volterra series predictor and a normalised RBF (NRBF) predictor are shown in Figure 3, using validation data sets (see *section 3.5*).

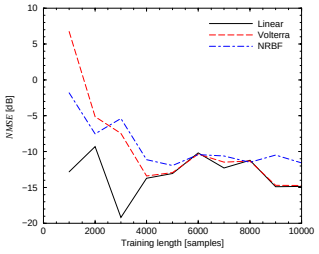


Figure 3: Validation data set *NMSE* vs training length for range cell 14 of the $12ms^{-1}$ data set. *Results are shown for a 10 tap linear predictor, a cubic (order=3) Volterra series predictor with an embedding dimension of 10, and a NRBF with an embedding dimension of 12, and 100 centres. An embedding delay of 1 sample was used by all predictors.*

What is immediately noticeable from Figure 3 is that the simple 10 tap linear predictor always does at least as well as the nonlinear predictors. This suggests that the predictor function for this sea clutter data set is linear. The cubic Volterra series predictor converges upon the linear solution at around about a training length of 4000 samples. The NRBF predictor displays difficulty in approximating the linear predictor function, showing approximately a $4dB$ performance loss relative to the linear predictor after a training length of $10,000$ samples.

The effect of increasing the embedding delay for each predictor is shown in Figure 4. As can be seen, increasing the embedding delay does not improve the prediction perform-

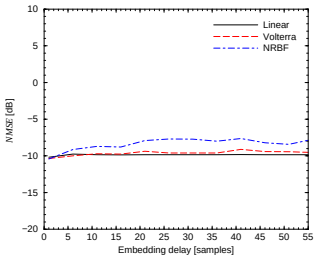


Figure 4: Validation data set *NMSE* vs embedding delay for range cell 14 of the $12ms^{-1}$ data set. *Results are shown for a 10 tap linear predictor, a cubic (order=3) Volterra series predictor with an embedding dimension of 10, and a NRBF with an embedding dimension of 12 and 100 centres. A training length of 6000 samples was used by all predictors.*

ance of any of the predictors.

4.2 Prediction performance for all 32 range cells in the $12\mathrm{ms}^{-1}$ data set

The results obtained in Figures 2 to 4 are representative of the results found for the other range cells in the $12ms^{-1}$ data set. This is illustrated in Figure 5 which displays prediction performance versus range cell for the $12ms^{-1}$ data set.

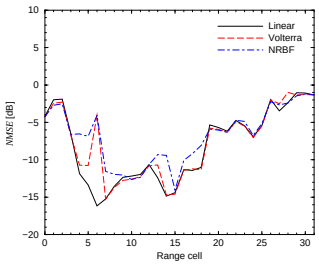


Figure 5: Validation data set *NMSE* vs range cell for the $12ms^{-1}$ data set. *Results are shown for a 10 tap linear predictor, a cubic (order=3) Volterra series predictor with an embedding dimension of 10, and a NRBF with an embedding dimension of 20 and 100 centres. An embedding delay of 1 sample and a training length of 10,000 samples were used by all the predictors.*

4.3 Prediction for all 32 range cells in each wind speed data set

The results in *sections 4.1* and *4.2* suggest that the predictor function of the $12ms^{-1}$ data is linear. In fact, this result was found for each of the different wind speed data sets. Additionally, it was found that, in general, the clutter samples collected during higher wind speeds were more predictable than those collected during lower wind speeds. However, there was also found to be a variation in predictability across range cells, which appeared to be dependent on the number and distribution of clutter amplitude spikes. Figure 6 shows the results for a linear predictor for all 32 range cells of several wind speed data sets.

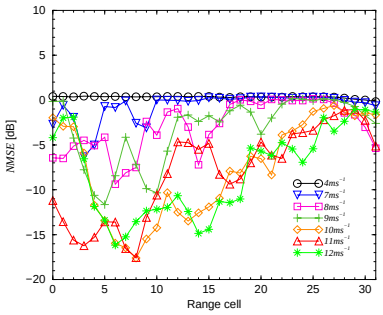


Figure 6: Validation data set *NMSE* vs range cell for several wind speed data sets. *Results are shown for a 10 tap linear predictor which used a training length of 10,000 samples and an embedding delay of 1 sample.*

5 Prediction of the Dawber data

As in *section 4.1*, the criteria discussed in *section 3.3* were used as a starting point to choose the NRBF embedding dimension and embedding delay. For the Dawber-VV data a value of $D_{ML} = 3.48$ was estimated, and an embedding dimension of 10 was chosen. For the Dawber-HH data a value of $D_{ML} = 7.57$ was estimated and an embedding dimension of 17 was selected. A plot of the mutual information of both Dawber data sets is given in Figure 7. There are no obvious minima in the mutual information plots. An embedding delay of 1 sample, for each data set, was therefore selected as a starting point.

Figure 8 shows the prediction performance of the nonlinear and linear predictors on both the Dawber data sets. As was found in *section 4* of this paper for the wavetank data, the predictor function of the Dawber data sets appears to be linear. Interestingly, the Dawber-VV data is more predictable than the Dawber-HH data despite the fact that it

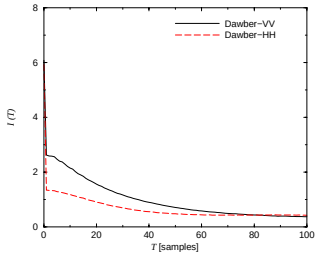


Figure 7: Mutual information plot for Dawber data sets.

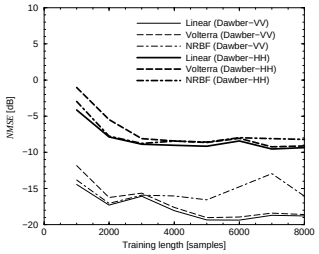


Figure 8: Validation data set *NMSE* vs training length for Dawber-VV (thin lines), and Dawber-HH (thick lines) data sets. *A linear predictor with 10 taps, and a cubic (order=3) Volterra series predictor with an embedding dimension of 10 were used on both Dawber data sets. A NRBF with an embedding dimension of 10 and 100 centres was used on the Dawber-VV data. A NRBF with an embedding dimension of 17 and 100 centres was used on the Dawber-HH data. An embedding delay of 1 sample was used by all the predictors.*

was collected during a lower wind speed. This is an area of ongoing research.

Figure 9 shows that increasing the embedding delay does not improve the prediction performance of the linear predictor or of the nonlinear predictors for either Dawber data set.

6 Detection strategies

A fixed threshold detector and a linear predictor-detector have been used to compare the performance of the nonlinear predictor-detectors.

6.1 False alarm rate and detector design

A false alarm refers to noise or clutter being mistaken for a target signal. In a radar detector design, the aim is to fix the probability of false alarm, P_{fa}, and maximise the probability of detection, P_d. The ideal situation would be if $P_{fa} = 0$,

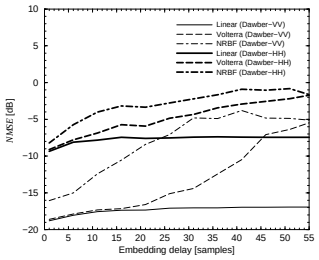


Figure 9: Validation data set *NMSE* vs embedding delay for Dawber-VV (thin lines), and Dawber-HH (thick lines) data sets. *A linear predictor with 10 taps, and a cubic (order=3) Volterra series predictor with embedding dimension of 10 were used on both Dawber data sets. A NRBF with an embedding dimension of 10 and 100 centres was used on the Dawber-VV data. A NRBF with an embedding dimension of 17 and 100 centres was used on the Dawber-HH data. A training length of 8000 samples was used by all the predictors.*

and $P_d = 1$. This would mean no false alarms, and if a target were present, it would always be detected. Receiver operating curves (plots of P_{fa} versus P_d) can be used to assess detection performance.

6.2 Fixed threshold detector

A threshold was set for this detector using a small set of clutter-only data to form a clutter amplitude histogram. The histogram was used to determine a threshold level that would result in a tolerable number of false alarms.

6.3 Predictor-detectors

The predictor (linear, RBF, or Volterra series) was trained using a small set of clutter-only data. After training, the free parameters of the predictor were fixed. Another small set of clutter-only data was then passed through the predictor, and the errors produced were used to form a histogram, which was used to set an error threshold for a desired P_{fa}. The idea is that during training the predictor should learn to recognise the clutter. Therefore, if clutter-only data is present at the input, the predictor will produce a small error. If a target as well as clutter is present at the input, then the predictor will produce a large error.

7 Detection results

As the prediction results for both the wavetank and Dawber data sets have been shown to have linear predictor functions, detection results are only presented for the wavetank data. Figures 10 and 11 show receiver operating curves (ROC's)

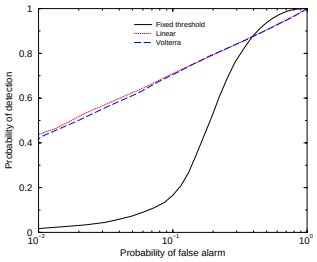


Figure 10: Detection of $12ms^{-1}$ wavetank data.

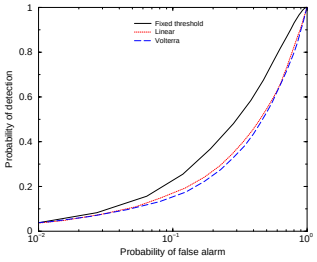


Figure 11: Detection of $4ms^{-1}$ wavetank data.

for range cell 16 of the $12ms^{-1}$ and $4ms^{-1}$ data sets, respectively. In each case a Swerling [12] (fluctuating Rayleigh) target was used, and the signal to clutter ratio was set to $0.2dB$. Small signal to clutter ratios are of particular interest in practice. In both plots an embedding dimension of 10, an embedding delay of 1, and a training length of 6000 samples were used for both the Volterra, and the linear predictor-detectors.

As expected from the prediction results in *section 4*, the linear and Volterra series predictor-detector results are very similar. The predictor-detectors perform better than the fixed threshold detector, (for low probabilities of false alarm), on the $12ms^{-1}$ data, which has a correlation length of approximately 170 samples. The fixed threshold detector performs better than the predictor-detectors on the $4ms^{-1}$ data, which is uncorrelated from sample to sample.

8 Conclusions

Prediction results have been presented for new non-coherent sea clutter data sets, which show no evidence of nonlinear predictability. However, the pulse compression used in the collection of the wavetank data, and the fairly low resolution of the Dawber data could be responsible for the lack of nonlinear predictability. Therefore, whilst the use of nonlinear predictor-detectors could not be justified to improve the performance of a radar detector for any of the clutter data sets analysed in this paper, this is not to say that this is the case for sea clutter in general.

Linear predictor-detectors may be used to improve detection performance, relative to a fixed threshold detector, as long as the clutter data concerned has a sufficiently long correlation length.

9 Acknowledgements

The authors of this paper would like to express their gratitude to the Defence Evaluation and Research Agency, UK, for providing the wavetank and Dawber sea clutter data sets. The software to calculate the maximum likelihood correlation dimension, and average mutual information, was provided by Dr. Charles Unsworth.

References

[1] K. D. Ward, "Compound representation of high resolution sea clutter," *IEE Electronics Letters*, vol. 17, no. 16, pp. 561–563, 1981.

[2] H. Leung and S. Haykin, "Is there a radar clutter attractor?," *Applied Physics Letters*, vol. 56, no. 6, 1990.

[3] S. Haykin and B. X. Li, "Detection of signals in chaos," *Proceedings of the IEEE*, vol. 83, pp. 95–122, January 1995.

[4] S. Haykin and S. Puthusserypady, "Chaotic dynamics of sea clutter," *Choas*, vol. 7, no. 4, pp. 777–802, 1997.

[5] M. Davies, "Looking for non-linearities in sea clutter," in *IEE Radar and Sonar Signal Processing*, (Peebles, Scotland), July 1998.

[6] J. L. Noga, *Bayesian state-space modelling of spatio-temporal non-Gaussian radar returns*. PhD thesis, Cambridge University, UK, 1998.

[7] M. Benaim, "On functional approximation with normalized Gaussian units," *Neural Computation*, vol. 6, pp. 319–333, 1994.

[8] S. Haykin, "Making sense of a complex world," *IEEE Signal Processing Magazine*, vol. 15, pp. 66–81, May 1998.

[9] J. C. Schouten, F. Takens, and C. M. van den Bleek, "Estimation of the dimension of a noisy attractor," *Physical Review E*, vol. 50, pp. 1851–1861, September 1994.

[10] H. D. I. Abarbanel, *Analysis of observed chaotic data*. Springer-Verlag, 1996.

[11] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, pp. 1134–1140, February 1986.

[12] M. I. Skolnik, *Introduction to radar systems*. McGraw-Hill, 1981.

References

[1] A. Hero, "Highlights of statistical signal and array processing," *IEEE Signal Processing Magazine*, vol. 15, pp. 21–64, September 1998.

[2] J. Gleick, *Chaos*, ch. "Chaos and beyond". Abacus, 1987.

[3] T. S. Parker and L. O. Chua, "Chaos: A tutorial for engineers," *Proceedings of the IEEE*, vol. 75, pp. 982–1008, August 1987.

[4] P. E. Strauch, *Nonlinear noise cancellation*. PhD thesis, University of Edinburgh, May 1997.

[5] K. D. Ward, C. J. Baker, and S. Watts, "Maritime surveillance radar part 1: radar scattering from the ocean surface," *IEE Proceedings Part F.*, vol. 137, pp. 51–62, April 1990.

[6] S. Watts, C. J. Baker, and K. D. Ward, "Maritime surveillance radar part 2: detection performance prediction in sea clutter," *IEE Proceedings Part F.*, vol. 137, pp. 63–72, April 1990.

[7] M. I. Skolnik, *Introduction to radar systems*. McGraw-Hill, second edition ed., 1981.

[8] K. D. Ward and S. Watts, "Radar sea clutter," *Microwave Journal*, vol. 28, pp. 109–121, June 1985.

[9] S. Haykin and H. Leung, "Chaotic model of sea clutter using a neural network," in *Proceedings of SPIE- third international society for optical engineering*, (San Diego, CA, USA), pp. 18–21, 8-12 August 1989.

[10] H. Leung and S. Haykin, "Is there a radar clutter attractor?," *Applied Physics Letters*, vol. 56, pp. 593–595, February 1990.

[11] N. He and S. Haykin, "Chaotic modelling of sea clutter," *Electronics Letters*, vol. 28, pp. 2076–2077, October 1992.

[12] H. Leung and T. Lo, "Chaotic radar signal processing over the sea," *IEEE Journal Of Oceanic Engineering*, vol. 18, pp. 287–295, July 1993.

[13] B. X. Li and S. Haykin, "Chaotic characterisation of sea clutter," *L'Onde Electrique*, vol. 74, May-June 1994.

[14] S. Haykin, "Chaotic characterisation of sea clutter: new experimental results and novel applications," in *Proceedings of the 29th Asilomar conference on signals systems and computers*, (Monterey, CA, USA), pp. 1076–1080, October 1995.

[15] H. Leung, "Applying chaos to radar detection in an ocean environment: an experimental study," *IEEE Journal of Oceanic Engineering*, vol. 20, pp. 56–64, January 1995.

[16] H. Leung, "An intelligent radar recognition system for surveillance," in *IEEE International Conference on Systems, Man, and Cybernetics: Intelligent Systems for the 21st Century*, vol. 3, (Vancouver, BC, Canada), pp. 2280–2285, 22-25 October 1995.

[17] S. Haykin and X. B. Li, "Detection of signals in chaos," *Proceedings of the IEEE*, vol. 83, pp. 95–122, January 1995.

[18] S. Haykin, "Neural networks expand SP's horizons," *IEEE Signal Processing Magazine*, vol. 13, pp. 24–49, March 1996.

[19] S. Haykin and S. Puthusserypady, "Chaotic dynamics of sea clutter: an experimental study," in *Proceedings of IEE Radar '97*, (Edinburgh, Scotland, UK), pp. 75–79, 14-16 October 1997.

[20] S. Haykin and S. Puthusserypady, "Chaos, sea clutter, and neural networks," in *31st Asilomar conference on signals, systems and computers*, vol. 2, (Pacific Grove, CA, USA), pp. 1224–1227, 2-5 November 1997.

[21] S. Haykin and S. Puthusserypady, "Chaotic dynamics of sea clutter," *Chaos*, vol. 7, no. 4, pp. 777–802, 1997.

[22] K. D. Ward, "Compound representation of high resolution sea clutter," *Electronics Letters*, vol. 17, pp. 561–563, August 1981 1981.

[23] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, pp. 10–26, July 1991.

[24] S. Haykin, *Neural Networks: A comprhensive foundation*. Prentice-Hall, 1994.

[25] D. S. Broomhead, J. P. Huke, and M. Potts, "Cancelling deterministic noise by constructing non-linear inverses to linear filters," *Physica D*, vol. 89, pp. 439–458, 1996.

[26] W. M. Denny, "K-distributed sea clutter: performance predictions made easy," in *Proceedings of IEE Radar '97*, (Edinburgh, Scotland, UK), pp. 209–213, 14-16 October 1997.

[27] A. Palmer, R. Kropfli, and C. Fairall, "Signatures of deterministic chaos in radar sea clutter and ocean surface winds," *Chaos*, vol. 5, no. 3, pp. 613–617, 1995.

[28] G. Kubin, "What is a chaotic signal?," in *IEEE Workshop on nonlinear signal and image processing*, vol. 1, (Halkdiki, Greece), pp. 141–144, 20-22 June 1995.

[29] S. Haykin and J. Principe, "Making sense of a complex world," *IEEE Signal Processing Magazine*, vol. 15, pp. 66–81, May 1998.

[30] F. Takens, *Dynamical Systems and Turbulence, Warwick 1980. Lecture notes in Mathematics*, vol. 898, ch. On the numerical determination of the dimension of an attractor. Springer-Verlag, 1981.

[31] K. T. Alligood, T. D. Sauer, and J. A. York, *Chaos: An introduction to dynamical systems*. Springer-Verlag, 1997.

[32] H. D. I. Abarbanel, *Analysis of observed chaotic data*. Springer-Verlag, 1996.

[33] D. Brook and R. J. Wynne, *Signal Processing Principles and Applications*. Edward Arnold, 1988.

[34] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, pp. 1134–1140, February 1986.

[35] J. C. Schouten, F. Takens, and C. M. van den Bleek, "Estimation of the dimension of a noisy attractor," *Physical Review E*, vol. 50, pp. 1851–1861, September 1994.

[36] E. Ott, T. Sauer, and J. A. Yorke, eds., *Coping with chaos: analysis of chaotic data and the exploitation of chaotic systems*. Wiley, 1994.

[37] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D*, vol. 16, pp. 285–317, 1985.

[38] M. Davies, "Looking for non-linearities in sea clutter," in *IEE Radar and Sonar Signal Processing*, (Peebles, Scotland, UK), pp. 12/1–12/2, July 1998.

[39] J. L. Noga, *Bayesian state-space modelling of spatio-temporal non-Gaussian radar returns*. PhD thesis, Cambridge University, UK, 1998.

[40] G. V. Trunk, "Radar properties of non-rayleigh sea clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 8, pp. 196–204, March 1972.

[41] P. H. Y. Lee, J. D. Barter, K. L. Beach, C. L. Hindman, B. M. Lake, H. Rungaldier, J. C. Shelton, A. B. Williams, R. Yee, and H. C. Yuen, "X-band microwave backscattering from ocean waves," *Journal of Geophysical Research*, vol. 100, pp. 2591–2611, February 1995.

[42] P. H. Y. Lee, J. D. Barter, K. L. Beach, E. Caponi, C. L. Hindman, B. M. Lake, H. Rungaldier, and J. C. Shelton, "Power spectral lineshapes of microwave radiation backscattered from sea surfaces at small grazing angles," *IEE Proceedings Radar, Sonar Navig.*, vol. 142, pp. 252–258, October 1995.

[43] F. E. Nathanson, *Radar design principles*. McGraw-Hill, 2nd ed., 1991.

[44] A. Farina and P. Lombardo, "Modelling of a mixture of K-distributed and Gaussian clutter for coherent radar detection," *Electronics Letters*, vol. 30, pp. 520–521, March 1994.

[45] S. Watts, "Radar detection prediction in K-distributed sea clutter and thermal noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, pp. 40–45, January 1987.

[46] C. J. Baker and K. D. Ward, "I-band multipath propagation over the sea surface," in *AGARD Conference No.419 (Scattering and propagation in random media*, (Neuilly Surseine, France), pp. 25/1 – 25/4, 1987.

[47] A. Farina, A. Russo, and F. A. Studer, "Coherent radar detection in log-normal clutter," *IEE Proceedings Part F.*, vol. 133, pp. 39–54, February 1986.

[48] G. R. Cooper and C. D. McGillem, *Probabilistic methods of signal and system analysis*. HRW International Editions, 2nd ed., 1986.

[49] T. Hair, T. Lee, and C. J. Baker, "Statistical properties of multi-frequency high-range-resolution sea reflections," *IEE Proceedings Part F*, vol. 138, pp. 75–79, April 1991.

[50] P. Embrechts, C. Klueppelberg, and T. Mikosch, *Modelling extreme events*. Springer-Verlag, 1997.

[51] N. A. J. Hastings and J. B. Peacock, *Statistical distributions*. Butterworths, 1975.

[52] G. V. Trunk and S. F. George, "Detection of targets in non-Gaussian sea clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 6, pp. 620–628, September 1970.

[53] R. Valenzuela and M. B. Laing, "On the statistics of sea clutter," tech. rep., Naval Research Laboratory, Washington DC, USA, December 1971.

[54] D. C. Schleher, "Radar detection in weibull clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 12, pp. 736–743, November 1976.

[55] F. A. Fay, J. Clarke, and R. S. Peters, "Weibull distribution applied to sea clutter," in *IEE International Conference Radar '77*, (London, UK), pp. 101–104, 25-28 October 1977.

[56] E. Jakeman and P. N. Pusey, "A model for non-rayleigh sea echo," *IEEE Transactions on Antennas and Propagation*, vol. 24, pp. 806–814, November 1976.

[57] E. Jakeman and P. N. Pusey, "Statistics of non-rayleigh microwave sea echo," in *IEE International Conference Radar '77*, (London, UK), pp. 105–109, 25-28 October 1977.

[58] T. J. Nohara, S. Haykin, B. W. Currie, and C. Krasnor, "Towards the improved detection of small ice targets in K-distributed sea clutter," in *Proceedings of the 1989 international symposium on noise and clutter rejection in radars and imaging sensors*, (Kyoto, Japan), pp. 66–71, 1989.

[59] S. Haykin, C. Krasnor, T. J. Nohara, B. W. Currie, and D. Hamburger, "A coherent dual-polarised radar for studying the ocean environment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 29, pp. 189–191, January 1991.

[60] T. J. Nohara and S. Haykin, "Canadian east coast radar trials and the K-distribution," *IEE Proceedings Part F*, vol. 138, pp. 80–88, April 1991.

[61] A. Farina, F. Gini, M. V. Greco, and L. Verrazzini, "Analysis of sea clutter radar data," in *CIE International Conference of Radar Proceedings*, (Beijing, China), pp. 115–118, October 1996.

[62] A. Farina, F. Gini, M. V. Greco, and L. Verrazzani, "High resolution sea clutter data: statistical analysis of recorded live data," *IEE Proceedings Radar, Sonar Navig.*, vol. 144, pp. 121–130, June 1997.

[63] C. Bouvier, L. Martinet, G. Favier, and M. Artaud, "Simulation of radar sea clutter using autoregressive modelling and K-distribution," in *IEEE International Radar Conference 1995*, (Alexandria, VA, USA), pp. 425–430, 8-11 May 1995.

[64] H. C. Chan, "Radar sea clutter at low grazing angles," *IEE Proceedings Part F*, vol. 137, pp. 102–112, April 1990.

[65] S. Watts, "A practical approach to the prediction and assessment of radar performance in sea clutter," in *IEEE International Radar Conference 1995*, (VA, USA), pp. 181–186, 8-11 May 1995 1995.

[66] S. Watts, "Cell-averaging CFAR gain in spatially correlated K-distributed clutter," *IEE Proceedings on Radar, Sonar Navig.*, vol. 143, pp. 321–327, October 1996.

[67] W. M. Denny, "Sea clutter – compound k distribution," Tech. Rep. Technical note: K-clutter No.1, G. E. C. Marconi Avionics, January 1995.

[68] J. K. Jao, "Amplitude distribution of composite terrain radar clutter and the K-distribution," *IEEE Transactions on Antennas and Propagation*, vol. 32, pp. 1049–1062, October 1984.

[69] C. J. Oliver, "The representation of correlated clutter textures in coherent images," *Inverse Problems*, vol. 4, no. 3, pp. 843–866, 1988.

[70] E. Jakeman, "On the statistics of K-distributed noise," *Journal of Physics A*, vol. 13, pp. 31–48, 1980.

[71] J. A. Ritcey and P. K. Tran, "Tail behavior of compound clutter models," in *Princeton workshop on Information and System Sciences*, 1992.

[72] V. Anastassopoulos and G. A. Lampropoulos, "A generalized compound model for radar clutter," in *IEEE National Radar Conference 1994*, (Atlanta, GA, USA), pp. 41–45, 29-31 March 1994.

[73] C. J. Baker, K. D. Ward, and S. Watts, "The significance and scope of the compound K-distribution model for sea clutter," in *IEE International Radar Conference 1987*, (London, UK), pp. 207–211, 19-21 October 1987.

[74] S. Watts and K. D. Ward, "Spatial correlation in K-distributed sea clutter," *IEE Proceedings Part F.*, vol. 134, pp. 526–532, October 1987.

[75] M. Sekine, T. Musha, Y. Tomita, T. Hagisawa, T. Irabu, and E. Kiuchi, "Log-weibull distributed sea clutter," *IEE Proceedings Part F.*, vol. 127, pp. 225–228, June 1980.

[76] R. D. Pierce, "Application of the positive alpha-stable distribution," in *IEEE Signal Processing Workshop on Higher-Order Statistics*, (Banff, Alberta, Canada), pp. 420–424, 21-23 July 1997.

[77] J. Ilow and H. Leung, "No evidence of stable distributions in radar clutter," in *IEEE Signal Processing Workshop on Higher-Order Statistics*, (Banff, Alberta, Canada), pp. 264–267, 21-23 July 1997.

[78] E. Conte, G. Galati, and M. Longo, "Exogenous modelling of non-Gaussian clutter," *Journal of the Institution of Electronic and Radio Engineers*, vol. 57, pp. 151–155, July/August 1987.

[79] E. Conte and M. Longo, "characterisation of radar clutter as a spherically invariant random process," *IEE Proceedings Part F.*, vol. 134, pp. 191–197, April 1987.

[80] E. Conte, M. Longo, and M. Lops, "Modelling and simulation of non-Rayleigh radar clutter," *IEE Proceedings Part F.*, vol. 138, pp. 121–130, April 1991.

[81] M. Rangaswamy, *Spherically invariant random processes for radar clutter modeling, simulation and distribution identification*. PhD thesis, Syracuse University, 1992.

[82] J. L. Noga and W. J. Fitzgerald, "Modelling sea clutter using conditional heteroscedastic models," in *Proceedings of EUSIPCO '98*, vol. 3, (Rhodes, Greece), pp. 1361–1364, 8-11 September 1998.

[83] T. Azzarelli, "General class of non-Gaussian coherent clutter models," *IEE Proceedings Radar, Sonar Navig.*, vol. 142, pp. 61–70, April 1995.

[84] M. Rangaswamy, D. Weiner, and A. Ozturk, "Computer generation of correlated non-gaussian radar clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 106–116, January 1995.

[85] D. Blacknell, "New method for the simulation of correlated K-distributed clutter," *IEE Proceedings Radar, Sonar Navig.*, vol. 141, pp. 53–58, February 1994.

[86] D. Blacknell, A. Blake, P. Lombardo, and C. J. Oliver, "A comparison of simulation techniques for correlated gamma and K-distributed images for SAR applications," in *International Geoscience and Remote Sensing Symposium 1994*, vol. 4, (Pasadena, CA, USA), pp. 2182–2184, 8-12 August 1994.

[87] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, 2nd ed., 1992.

[88] B. Liu and D. C. Munson, "Generation of a random sequence having a jointly specified marginal distribution and autocovariance," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, pp. 973–983, December 1982.

[89] K. Yao, "A representation theorem and its applications to spherically-invariant random processes," *IEEE Transactions on Informatioin Theory*, vol. 19, pp. 600–608, September 1973.

[90] C. J. Oliver and R. J. A. Tough, "On the simulation of correlated K-distributed random clutter," *Otica Acta*, vol. 33, no. 3, pp. 223–250, 1986.

[91] B. C. Armstrong and H. D. Griffiths, "Modelling spatially correlated K-distributed clutter," *Electronics Letters*, vol. 27, pp. 1355–1356, July 1991.

[92] J. G. Proakis, *Digital Communications*. McGraw-Hill, 3rd ed., 1995.

[93] L. J. Marier, "Correlated K-distributed clutter generation for radar detection and track," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 568–580, April 1995.

[94] R. J. A. Tough and K. D. Ward, "The generation of correlated K-distributed noise," Tech. Rep. DRA/CIS/CBC3/WP94001/2.0, Defence Research Agency, Farnborough, Hampshire, England, GU14 6TD, April 1994.

[95] R. Tough and K. Ward, "The correlation properties of gamma and other non-gaussian processes generated by memoryless nonlinear transformation," *submitted to Journal of Physics D.*, 1999.

[96] M. Abramowitz and I. A. Stegun, eds., *Handbook of mathematical functions*. United States Department of Commerce and National Bureau of Standards, 1966.

[97] A. Heck, *Introduction to Maple*. Springer-Verlag, 2nd ed., 1996.

[98] E. Kreyszig, *Advanced Engineering Mathematics*. Wiley, 7th ed., 1993.

[99] I. Scott and B. Mulgrew, "Nonlinear system identification and prediction using orthonormal functions," *IEEE Transactions on signal processing*, vol. 45, pp. 1842–1853, July 1997.

[100] M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D*, vol. 35, pp. 335–356, 1989.

[101] M. Schetzen, "Nonlinear system modelling based on the Wiener theory," *Proceedings of the IEEE*, vol. 69, pp. 1557–1573, December 1981.

[102] I. W. Sandberg, "The mathematical foundations of associated expansions for mildly nonlinear systems," *IEEE Transactions on Circuits and Systems*, vol. 30, pp. 441–454, July 1983.

[103] J. Lee and V. J. Mathews, "A fast recursive least squares adaptive second-order volterra filter and its performance analysis," *IEEE Transactions on Signal Processing*, vol. 41, pp. 1087–1101, March 1993.

[104] K. C. Nisbet, "Reduced state methods in nonlinear prediction," *Signal Processing*, vol. 48, pp. 37–49, 1996.

[105] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[106] T. Poggio, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, September 1990.

[107] S. Chen, A. Billings, C. F. N. Cowan, and P. M. Grant, "Nonlinear systems identification using radial basis functions," *International Journal of Systems Science*, vol. 21, pp. 2513–2539, December 1990.

[108] J. A. Leonard and M. A. Kramer, "Radial basis function networks for classifying process faults," *IEEE Control Systems Magazine*, vol. 11, pp. 31–38, April 1991.

[109] 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, pp. 302–309, March 1991.

[110] A. Sherstinsky and R. W. Picard, "On training Gaussian radial basis functions for image coding," Vision and Modelling Group Technical Report 188, MIT Media Laboratory, MIT Media Laboratory, E15-383, 20 Ames Street, Cambridge, MA, USA., February 1992.

[111] F. Miguel and A. Acosta, "Radial basis function and related models: An overview," *Signal Processing*, vol. 45, pp. 37–58, 1995.

[112] M. J. L. Orr, "Introduction to radial basis function networks," 1996. http://www.anc.ed.ac.uk/ mjo/intro/intro.html.

[113] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Processing Magazine*, vol. 13, pp. 50–65, March 1996.

[114] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1996.

[115] A. O. Steinhardt, "Householder transforms in signal processing," *IEEE Acoustics, Speech and Signal Processing*, vol. 5, pp. 4–12, July 1988.

[116] S. Haykin, *Adaptive filter theory*. Prentice Hall, 3rd ed., 1996.

[117] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[118] H. W. Werntges, "Partitions of unity to improve neural function approximators," in *IEEE International Conference on Neural Networks*, vol. 2, (San Francisco, CA, USA), pp. 914–918, 28 March - 1 April 1993.

[119] M. Benaim, "On the functional approximation with normalized Gaussian units," *Neural Computation*, vol. 6, pp. 314–333, 1994.

[120] L. Xu, A. Kryzyzak, and A. Yuille, "On radial basis function nets and kernel regression: statistical consistency, convergence rates, and receptive field size," *Neural Networks*, vol. 7, no. 4, pp. 609–628, 1994.

[121] I. Cha and S. A. Kassam, "Interference cancellation uding radial basis function networks," *Signal Processing*, vol. 47, pp. 247–268, 1995.

[122] R. Shorten and R. Murray-Smith, "Side effects of normalising radial basis function networks," *International Journal of Neural Systems*, vol. 7, pp. 167–179, May 1996.

[123] G. Bugmann, "Normalized Gaussian radial basis function networks," *Neurocomputing*, vol. 20, no. 1-3, pp. 97–110, 1998.

[124] C. Chinrungrueng and C. H. Sequin, "Optimal adaptive k-means algorithm with dynamic adjustment of learning rate," *IEEE Transactions on Neural Networks*, vol. 6, pp. 157–169, January 1995.

[125] S. Chen, "Nonlinear time series modelling and prediction using Gaussian rbf nerworks with enhanced clustering and rls learning," *Electronics Letters*, vol. 31, pp. 117–118, January 1995.

[126] J. L. Eaves and E. K. Reedy, *Principles of modern radar*. Van Nostrand Reinhold, 1987.

[127] B. Mulgrew, P. Grant, and J. Thompson, *Digital signal processing concepts and applications*. Macmillan Press LTD, 1999.

[128] S. Watts, "U.K. patent number 2191052," 1981. Racal Radar Defence Systems.

[129] G. Hansen, "Constant false alarm rate processing in search radars," in *IEE International Conference on Radar - Present and Future*, (London, U.K.), pp. 325–332, 25 October 1973.

[130] E. O. Lewis, B. W. Currie, and S. Haykin, *Detection and classification of ice*. Research Studies Press, 1987.

[131] V. Y. F. Li and K. M. Miller, "Target detection in radar: current status and future possibilities," *Journal of Navigation*, vol. 50, no. 2, pp. 303–313, 1997.

[132] P. P. Gandhi and S. A. Kassam, "Analysis of CFAR processors in nonhomogeneous background," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, pp. 427–445, July 1988.

[133] A. Papoulis, *Probability, random variables, and stochastic processes*. McGraw-Hill, 3rd ed., 1991.

[134] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge University Press, 1997.

[135] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. Macmillan, 1992.

[136] Racal radar defence systems limited, Wookey Hole Road, Wells, Somerset, BA5 1AA, England, *MIDAS technical description*, 1998.

[137] J. A. Rice, *Mathematical statistics and data analysis*. Wadsworth and Brooks/Cole, 1988.

[138] S. Watts, "Tutorial session 1b: Radar clutter and CFAR detection." Tutorial Session notes from IEE Conference Radar '97, October 1997.

[139] J. Daintith and R. D. Nelson, eds., *Dictionary of mathematics*. Penguin, 1989.