
Nonlinear Noise Cancellation

Paul E STRAUCH



A thesis submitted for the degree of Doctor of Philosophy.

The University of Edinburgh.

- May 1997 -

Abstract

Noise or interference is often assumed to be a random process. Conventional linear filtering, control or prediction techniques are used to cancel or reduce the noise. However, some noise processes have been shown to be nonlinear and deterministic. These nonlinear deterministic noise processes appear to be random when analysed with second order statistics. As nonlinear processes are widespread in nature it may be beneficial to exploit the coherence of the nonlinear deterministic noise with nonlinear filtering techniques.

The nonlinear deterministic noise processes used in this thesis are generated from nonlinear difference or differential equations which are derived from real world scenarios. Analysis tools from the theory of nonlinear dynamics are used to determine an appropriate sampling rate of the nonlinear deterministic noise processes and their embedding dimensions. Nonlinear models, such as the Volterra series filter and the radial basis function network are trained to model or predict the nonlinear deterministic noise process in order to reduce the noise in a system. The nonlinear models exploit the structure and determinism and, therefore, perform better than conventional linear techniques.

These nonlinear techniques are applied to cancel broadband nonlinear deterministic noise which corrupts a narrowband signal. An existing filter method is investigated and compared with standard linear techniques. A new filter method is devised to overcome the restrictions of the existing filter method. This method combines standard signal processing concepts (filterbanks and multirate sampling) with linear and nonlinear modelling techniques. It overcomes the restrictions associated with linear techniques and hence produces better performance. Other schemes for cancelling broadband noise are devised and investigated using quantisers and cascaded radial basis function networks. Finally, a scheme is devised which enables the detection of a signal of interest buried in heavy chaotic noise.

Active noise control is another application where the acoustic noise may be assumed to be a nonlinear deterministic process. One of the problems in active noise control is the inversion process of the transfer function of the loudspeaker. This transfer function may be nonminimum phase. Linear controllers only perform sub-optimally in modelling the non-causal inverse transfer function. To overcome this problem in conjunction with the assumption that the acoustic noise is nonlinear and deterministic a combined linear and nonlinear controller is devised. A mathematical expression for the combined controller is derived which consists of a linear system identification part and a nonlinear prediction part. The traditional filtered-x least mean squares scheme in active noise control does not allow the implementation of a nonlinear controller. Therefore, a control scheme is devised to allow a nonlinear controller in conjunction with an adaptive block least squares algorithm. Simulations demonstrate that the combined linear and nonlinear controller outperforms the conventional linear controller.

Declaration of originality

This thesis was composed entirely by myself. The work reported herein was conducted exclusively by myself in the Department of Electrical Engineering at the University of Edinburgh.

Paul E STRAUCH

May 1997

Acknowledgements

I would like to thank the following people for their invaluable assistance during the course of this PhD:

- Bernard Mulgrew, my supervisor, for his continuous support and guidance throughout my PhD. Also for reading and checking this thesis.
- Steve McLaughlin and Peter Grant for their support and guidance.
- Iain Scott, John Thompson, Justin Fackrell, Achilleas Stogioglou, Eng-Siong Chng, Yoo-sok Saw, Steve Bates, Bill Nailon, Richard Stirling-Gallacher, Ian Band and Jon Altuna for useful technical discussions and advice.
- The other members of the Signals and Systems Group for their support.
- My mother S. Strauch for providing financial support.

To my mother Sheila Strauch

Contents

List of Figures	vii
List of Tables	x
Abbreviations	xi
List of Symbols	xii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions of this Dissertation	2
1.3 Overview of the Thesis	3
2 Nonlinear Dynamics and Chaos	5
2.1 Introduction	5
2.2 History	10
2.3 Invariants of the Dynamics	11
2.3.1 Dimension	11
2.3.2 Lyapunov Exponents	13
2.4 Embedding Dimension	15
2.4.1 Taken's Delay Embedding Theorem	15
2.4.2 Singular System Analysis	16
2.4.3 False Nearest Neighbours	18
2.5 Sampling Rate	18
2.5.1 Average Mutual Information	18
2.6 Summary	20
3 Nonlinear Models	21
3.1 Introduction	21
3.2 Volterra Series Filter	22
3.3 Radial Basis Function Network	24
3.3.1 Topology	27
3.3.2 Estimation of the Basis Function Parameters	30
3.3.3 Training Algorithms for the Linear Weights	36
3.4 Chapter Summary	36
4 Interference Cancellation: Investigation of Broomhead's Filter Method	38
4.1 Introduction	38

4.2	Filter Method (Theory)	40
4.3	Theory: Simulations and Results	43
4.4	Practice: Simulations and Results	51
4.5	Comparison with Linear Techniques	52
4.5.1	Simulations and Results	53
4.6	Chapter Summary	56
5	Interference Cancellation: Modification and Re-embedding	58
5.1	Introduction	58
5.2	New Filter Method	58
5.2.1	FIR Filterbanks	60
5.2.2	Multirate Signal Processing	61
5.2.3	Adaptive Linear Filter	62
5.2.4	Nonlinear Model	63
5.2.5	Simulations and Results	63
5.2.6	Discussion	67
5.3	Quantisation	67
5.3.1	Design	68
5.3.2	Simulations and Results	69
5.3.3	Discussion	73
5.4	Cascaded Normalised Radial Basis Function Networks	74
5.4.1	Description of the Scheme	74
5.4.2	Simulations and Results	75
5.4.3	Discussion	77
5.5	Chapter Summary	77
6	Nonlinear Active Noise Control	79
6.1	Introduction	79
6.2	Background	82
6.2.1	Active Noise Control in Ducts	82
6.2.2	The Filtered-x Algorithm	84
6.2.3	Alternative Linear Control Schemes	87
6.3	Nonlinear Active Noise Control in a Duct	92
6.3.1	Introduction	92
6.3.2	Nonlinear Control of a Linear Plant	93
6.3.3	Development of a Combined Linear and Nonlinear Controller	94
6.3.4	Simulations and Results	98
6.3.5	Discussion	104
6.4	Chapter Summary	107
7	Conclusion	108
7.1	Review	108
7.2	Observations	110
7.3	Future Work	111

A Householder Transformation	112
B Original Publications	115
References	124

List of Figures

2.1	Aperiodic time series : (a) Time series 1 (b) Time series 2	6
2.2	PSD (averaged 50 times) : (a) Time series 1 (b) Time series 2	7
2.3	Autocorrelation (averaged 50 times) : (a) Time series 1 (b) Time series 2	7
2.4	Phase plane : (a) Time series 1 (b) Time series 2	8
2.5	Feigenbaum diagram of the Logistic map	9
2.6	Attractor of the Logistic map ($\mu = 4.0$)	9
2.7	The Lorenz attractor	10
2.8	Cantor set	12
2.9	Sensitivity to initial conditions	14
2.10	Singular System Analysis	17
2.11	Average Mutual Information $I(T)$ for Lorenz data	19
3.1	Second order Volterra filter with two delay elements	23
3.2	The radial basis function network	25
3.3	4 un-normalised Gaussian basis functions with 2 inputs: (a) $\sigma^2 = 5$ (b) $\sigma^2 = 50$	26
3.4	4 normalised Gaussian basis functions with 2 inputs : (a) $\sigma^2 = 5$ (b) $\sigma^2 = 50$	26
3.5	Early stopping	27
3.6	Early stopping : NMSE in dB for training, testing and validating	29
3.7	OLS : NMSE in dB for training, testing and validating	29
3.8	Logistic map (10000 training samples, $\alpha = 0.99$, 25 kernels) : (a) Initialisation (b) Final kernel locations	31
3.9	Lorenz (40000 training samples, $\alpha = 0.99$, 125 kernels) : (a) Initialisation (b) Final kernel locations	31
3.10	Mean square error of the variance	32
3.11	Logistic map with AWGN $\sim N(0, 0.025)$, 400 data points, 200 iterations, 25 kernels	33
3.12	Logistic map with AWGN $\sigma^2 = 0.025$, 400 data points, 200 iterations, 25 kernels	34
3.13	Lorenz, 4000 data points, 600 iterations, 125 kernels	34
3.14	Lorenz, 4000 data points, 600 iterations, 125 kernels	35
3.15	Logistic map : (a) Initialisation (b) Final kernel locations	35
3.16	Lorenz : (a) Initialisation (b) Final kernel locations	36
4.1	Example for a spectrum 'noise + signal of interest'	38
4.2	Nonlinear signal prediction	40
4.3	Filter method for separating signals	41
4.4	Delay embedding with a filtered time series	42
4.5	PSD : (a) Corrupted signal $y(n)$ (b) Filtered signal $u(n)$	44
4.6	Phase plane : (a) Original attractor (b) Filtered attractor	45
4.7	PSD of recovered signal of interest	45
4.8	NMSE in dB for training, testing and validating	46
4.9	Reconstructed attractor of the Ikeda map	46
4.10	PSD of signal of interest and Lorenz noise	47
4.11	PSD : (a) Corrupted signal $y(n)$ (b) Filtered signal $u(n)$	47
4.12	Time series : (a) Corrupted signal $y(n)$ and recovered signal $\hat{a}(n)$ (b) Demodulated signal	48
4.13	NMSE in dB for training, testing and validating	48
4.14	PSD : (a) Rössler chaos $a(n)$ (b) Filtered Lorenz + Rössler $u(n)$	49
4.15	NMSE in dB for training, testing and validating : (a) Training with the signal of interest present (b) Training without the signal of interest present	50
4.16	PSD of the recovered Rössler chaos	50

4.17	Attractors with different FIR filters : (a) 25 coefficients (b) 193 coefficients	51
4.18	PSD of the filtered time series $u(n)$ of the nonlinear model	52
4.19	PSD : (a) Output of the NRBF network (b) Recovered signal of interest	52
4.20	PSD of the recovered signal of interest	53
4.21	PSD : (a) Recovered signal of interest (b) Transfer function of the linear adaptive filter	54
4.22	Time series	54
4.23	Time series (a) Linear adaptive filter with 10 coefficients (b) Linear adaptive filter with 50 coefficients	55
4.24	PSD of the recovered Rössler chaos using a bandpass filter	55
4.25	PSD of the recovered Rössler chaos (a) Linear adaptive filter with 10 coefficients (b) Linear adaptive filter with 50 coefficients	56
5.1	Nonlinear interference cancellation	59
5.2	Sine wave corrupted by chaotic noise derived from the Lorenz system	60
5.3	PSD : (a) Lowpass filtered signal $v(n)$ (b) Bandpass filtered signal $d(n)$	60
5.4	Average mutual information $I(T)$	61
5.5	PSD : (a) Downsampled signal $v(nM)$ (b) Downsampled signal $d(nM)$	62
5.6	Interpolation and bandpass filtering	62
5.7	Error signal $e_1(nM)$ of the linear adaptive FIR filter	63
5.8	Linear filtering	64
5.9	Input and output SNR in dB with an embedding dimension of 2	64
5.10	Input and output SNR in dB with an embedding dimension of 3	65
5.11	Input and output SNR in dB with an embedding dimension of 4	65
5.12	Input and output SNR in dB with an embedding dimension of 5	66
5.13	PSD : Corrupted narrowband signal of interest in chaotic noise	66
5.14	Input and output SNR in dB for different FIR filter lengths	67
5.15	Nonlinear interference cancellation using a quantiser	68
5.16	PSD : Sine wave in chaotic noise	69
5.17	Time series : Signal $v(nM)$ and quantised signal (dotted) $v'(nM)$	69
5.18	PSD : (a) Signal $v(nM)$ (b) Quantised signal $v'(nM)$	70
5.19	PSD : (a) Signal $y(nM)$ without quantiser (b) Signal $y(nM)$ with quantiser	70
5.20	NMSE in dB for training, testing and validating : (a) Without quantiser (b) With quantiser	71
5.21	PSD : (a) Recovered signal of interest $\hat{a}(n)$ without quantisation (b) Recovered signal of interest $\hat{a}(n)$ with quantisation	71
5.22	PSD : Sine wave in chaotic noise	72
5.23	Nonlinear interference cancellation	72
5.24	PSD : (a) Signal $y(nM)$ (b) Recovered signal of interest $a'(nM)$	73
5.25	Cascaded NRBF networks for nonlinear interference cancellation	75
5.26	PSD : Three sine waves in chaotic noise	75
5.27	PSD : (a) Signal $y_1(n)$ (b) Signal $e_1(n)$	76
5.28	PSD : (a) Signal $y_2(n)$ (b) Signal $e_2(n)$	76
5.29	PSD : (a) Signal $y_3(n)$ (b) Signal $\hat{a}(n)$	77
6.1	Sound waves from out of phase operating acoustic sources (a) Low frequency - global destructive interference (b) High frequency - localised destructive (D) and constructive (C) interference	80
6.2	Active noise system using feedback control	81
6.3	Active noise system using feedforward control	81
6.4	Feedforward control in a duct	82
6.5	Block diagram of the ANC system with conventional adaptive FIR filter and feedback cancellation	83
6.6	Derivation of the filtered-x LMS algorithm : (a) noise canceller; (b) combining transfer functions; (c) adaptive structure	85
6.7	Filtered-x LMS, 20 filter coefficients, $\mu = 0.01$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB	86
6.8	Filtered-x RLS, 20 filter coefficients, $\alpha = 1.0$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB	87

6.9	Block diagram of the 'constrained filtered-x' LMS algorithm	88
6.10	Constrained filtered-x LMS, 20 filter coefficients, $\mu = 0.01$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) e(n) (b) NMSE / dB	89
6.11	Constrained filtered-x RLS, 20 filter coefficients, $\alpha = 1.0$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) e(n) (b) NMSE / dB	89
6.12	Modified filtered-x scheme with an additional 'secondary path' $\hat{H}_a(z)\hat{H}_e(z)$ and controller W	90
6.13	Modified filtered-x LMS, 20 filter coefficients, $\mu = 0.01$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) e(n) (b) NMSE / dB	91
6.14	Modified filtered-x RLS, 20 filter coefficients, $\alpha = 1.0$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) e(n) (b) NMSE / dB	91
6.15	Block diagram of the ANC system used in simulations	93
6.16	Block diagram of ANC system for derivation of the combined controller	94
6.17	NMSE in dB for 3 different controllers : $x(n) = 4x(n-1)(1-x(n-1))$, $M = 3$, $N = 10$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e = \hat{H}_e = 1$, $P(z) = 1$, $F(z) = 0$, $v1(n) = v2(n) = 0$, $m = 16$, 32 taps for $(\hat{H}_a\hat{H}_e)^{-1}$	98
6.18	NMSE in dB for 3 different controllers : $x(n) = 4x(n-1)(1-x(n-1))$, $M = 3$, $N = 10$, $H_a(z) = \hat{H}_a(z) = 1 + 0.5z^{-1}$, $H_e = \hat{H}_e = 1$, $P(z) = 1$, $F(z) = 0$, $v1(n) = v2(n) = 0$, $m = 0$, 32 taps for $(\hat{H}_a\hat{H}_e)^{-1}$	99
6.19	Attractor of the two dimensional stochastic nonlinear dynamical time series	100
6.20	The influence of the input signal $x(n)$, controller $H_a(z)^{-1}$ and the nature of the actuator $H_a(z)$ on the error signal $e(n)$	104
6.21	Duffing noise for $x(n)$, $M = 5$, $N = 5$, $H_a(z) = 1 + 0.5z^{-1}$, $H_e(z) = z^{-5}$, $m = 5$, 32 taps for $(\hat{H}_a\hat{H}_e)^{-1}$	105
6.22	Logistic noise for $x(n)$, $M = 5$, $N = 5$, $H_a(z) = 1 + 0.5z^{-1}$, $H_e(z) = z^{-5}$, $m = 5$, 32 taps for $(\hat{H}_a\hat{H}_e)^{-1}$	106
6.23	The m-step predictability of chaotic time series without feedback	107

List of Tables

4.1	SNR in dB for the different linear and nonlinear filters	56
6.1	NMSE in dB using a linear controller and stochastic noise for $x(n)$	102
6.2	NMSE in dB using a combined linear and nonlinear controller and stochastic noise for $x(n)$	102
6.3	NMSE in dB using a combined linear and nonlinear controller and stochastic noise for $x(n)$	102
6.4	NMSE in dB using a nonlinear controller and stochastic noise for $x(n)$	103
6.5	NMSE in dB using a nonlinear controller and stochastic noise for $x(n)$	103
6.6	NMSE in dB using a linear controller and nonlinear noise for $x(n)$	103
6.7	NMSE in dB using a nonlinear controller and nonlinear noise for $x(n)$	103
6.8	NMSE in dB using a nonlinear controller and nonlinear noise for $x(n)$	103
6.9	NMSE in dB using a combined linear and nonlinear controller and nonlinear noise for $x(n)$	104
6.10	NMSE in dB using a combined linear and nonlinear controller and nonlinear noise for $x(n)$	104

Abbreviations

ANC	Active noise control
AWGN	Additive white Gaussian noise
BP	Bandpass
DSP	Digital signal processor
EM	Expectation maximisation
FIR	Finite impulse response
HOS	Higher order statistics
i.i.d	Identical independent distributed
IIR	Infinite impulse response
LMS	Least means squares
LP	Lowpass
LS	Least squares
MIMO	Multiple input - multiple output
ML	Maximum likelihood
MLP	Multilayer perceptron
MMSE	Minimum mean square error
NN	Neural networks
MSE	Mean square error
NMSE	Normalised mean square error
NRBF	Normalised radial basis function
ODE	Ordinary differential equation
OLS	Orthogonal least squares
PCA	Principal component analysis
pdf	Probability density function
PSK	Phase shift key
PSD	Power spectral density

RBF	Radial basis function
RLS	Recursive least squares
SISO	Single input - single output
SNR	Signal to noise ratio
SVD	Singular value decomposition
VS	Volterra series

List of principal symbols

α	forgetting factor in recursive least squares algorithm
$a(n)$	signal of interest at time n
$d(n)$	desired signal at time n
$e(n)$	error signal at time n subjected to a minimisation process in the least squares sense
F	filter matrix
$F(z)$	transfer function of a feedback path
f_s	sampling frequency
$H_a(z)$	transfer function of a loudspeaker, driving unit, lowpass filter and D/A converter
$H_e(z)$	transfer function of an acoustic path, error microphone, lowpass filter and A/D converter
I	identity matrix
L	upsampling factor
M	downsampling factor
μ	step size parameter in least means squares algorithm
$P(z)$	transfer function of a plant
$p(\cdot \cdot)$	conditional probability
$P(j)$	prior probability of the j th kernel
$R_{xx}(l)$	autocorrelation function with lag l
σ^2	variance
Φ	nonlinear kernel
W	controller
$x(n)$	input signal at time n
x	input vector
X	input matrix
$\ \mathbf{x}\ $	Euclidean norm of vector \mathbf{x} , i.e. $\sqrt{\mathbf{x}^T \mathbf{x}}$.
$y(n)$	output signal at time n
ψ	embedding vector

$E\{\cdot\}$	expectation operator
$E[\cdot \cdot]$	conditional expectation operator
$N(0,1.0)$	normal distribution with zero mean and unit variance
$(\cdot)^+$	pseudo inverse operation
$(\cdot)^T$	transposition operation
$(\cdot)^{-1}$	inverse operation
*	convolution operation
$\hat{\cdot}$	estimated signal, value etc.

Chapter 1

Introduction

1.1 Motivation

Noise or interference is regarded generally as a nuisance in real life applications such as in systems and signal processing. Every day one experiences the negative effects of noise or interference, *e.g.* as background noise on a telephone line, blurred TV images, acoustic noise from engines or ventilation systems, speckle in ultra sound images, co-channel or intersymbol interference in mobile communications, sea clutter in radar signals etc. In some cases the noise or interference is so severe that it completely masks the signal of interest and may cause complete failure of an application. To cancel or to reduce noise or interference and, hence, to detect or to recover a signal of interest or to generate an acoustical noise-free zone are the two major objectives in this thesis.

The conventional approach for reducing noise or interference is based on linear signal processing techniques. Noise or interference in the past has been regarded as random processes. Usually these random processes are assumed to be white Gaussian or coloured. Linear techniques are then applied to average the white noise out, predict the signal of interest or use filtering techniques in the frequency domain. Filtering techniques are most straightforward and employ bandpass or lowpass filters to capture the bandlimited signal of interest. However, the linear filtering technique is not able to reduce the noise in the bandwidth of interest.

During the last three decades it has emerged that nonlinear deterministic systems can generate time series which have many properties of classical noise. This is not really surprising as most processes in nature are nonlinear. However, if the noise or interference is nonlinear and deterministic then it should be modelled as a nonlinear deterministic or even chaotic process rather than a stochastic one. Linear models exploit the incoherence of stochastic noise but are not able to exploit the coherence of nonlinear deterministic noise. Nonlinear models on the other hand, such as Volterra series (VS), radial basis function (RBF) networks or neural nets (NN), have been shown to be rather successful in modelling and predicting nonlinear deterministic time series.

A lot of research has been carried out (especially in NN) in devising fast, robust, stable and reliable learning methods and in different efficient network configurations. Although this is still an ongoing research topic,

there already exist a few efficient and straightforward nonlinear models which are easy to implement (*i.e.* VS filters and RBF networks).

Nonlinear models are usually far more complex and computational expensive than linear models. However, digital signal processors (DSP) are becoming much more faster and specialised (*e.g.* fuzzy or RBF chips), so that in the near future the implementation of nonlinear models will not cause major problems. For these reasons, it is beneficial to investigate the capabilities of nonlinear models to model or to predict nonlinear deterministic noise. It is very likely that nonlinear models will overcome a variety of problems which are encountered in linear filtering techniques (*e.g.* remaining noise in the bandwidth of interest or ill-conditioned inverse problems).

Two different applications are examined. These applications share the common theme that the noise which is corrupting a system or a signal of interest is nonlinear and deterministic. One application is the cancellation or reduction of noise or interference corrupting a narrowband signal of interest. The other one is cancelling acoustic noise actively from a fan or engine in a duct.

1.2 Contributions of this Dissertation

The work in this thesis deals with the cancellation or reduction of nonlinear deterministic noise or interference. Two different applications are investigated in conjunction with a variety of devised schemes to achieve nonlinear noise cancellation.

The first one is a filter method of Broomhead *et al.* [1] for cancelling nonlinear deterministic noise from a narrowband signal. This filter method is investigated and compared with conventional linear techniques. It is shown that the restrictions of the filter method are so severe that it will be impractical to implement. Further it is shown that if the same restrictive assumptions are fulfilled linear filtering techniques which are less computational expensive have the same or even better performance than the filter method.

To overcome those restrictions and limitations a new filter method is devised. The orthogonality property of the finite impulse response (FIR) filter which is the main restriction of [1] is removed. To compensate for the loss of the orthogonality property of the FIR filter filterbanks, decimators and an additional linear adaptive filter are implemented into the new filter method. The new filter method shows an improvement in performance (signal to noise ratio (SNR)) compared to linear filtering techniques.

A different filter method which uses cascaded RBF networks is designed and investigated. The initial idea of using cascaded RBF networks is shown. Unfortunately, simulations show that it is rather difficult to train the RBF networks in series. The whole scheme only performs sub-optimally and did not show any great improvement in performance compared with linear filtering techniques.

Another filter method to overcome the restrictions and limitations of Broomhead's filter method is devised.

It uses filterbanks, a quantiser and a nonlinear model. Simulations show that the quantiser successfully masks the signal of interest and the nonlinear model is not able to detect and try to model it. However, the additional quantisation noise makes it very difficult for the nonlinear model to estimate the nonlinear deterministic noise accurately. Therefore, this scheme performs only sub-optimally compared with linear filtering techniques.

Finally a filter method is devised which recovers and detects signals of interest in strong nonlinear deterministic noise. The filter method shows great improvement in performance compared with linear filtering techniques.

The second application is active noise control (ANC) in a duct. One of the problems in ANC is that the controller has to model the inverse of the loudspeaker. This loudspeaker may be nonminimum phase and, therefore, causes problems for a linear controller. To overcome this problem a combined linear and nonlinear controller is devised and implemented into an ANC scheme. It is shown that the traditional filtered-x LMS scheme is not valid with a nonlinear controller. Other alternative schemes are investigated in order to use a nonlinear controller. A new scheme is designed which enables the controller to be adapted by a block least squares technique. This scheme is also more efficient in the number of additional models than the presented alternative schemes. A mathematical model is derived for the combined linear and nonlinear controller and is verified in a variety of different simulations. The combined linear and nonlinear controller shows a great increase in performance in comparison with nonlinear or linear models.

1.3 Overview of the Thesis

Chapter 2 follows this introduction which provides some background information on nonlinear dynamics and chaos. Firstly, it is explained what it is meant when a process is nonlinear and deterministic. The ability of chaotic processes to masquerade as random processes when analysed with second order statistics is highlighted. Different measures which are invariant to initial conditions and determine the nonlinear dynamics are shown. As this thesis is dealing with time series observations of noisy signals in general the emphasis is put on the estimation of the sampling rate and the embedding dimension. The sampling rate and the embedding dimension are two important quantities which enable nonlinear models to exploit the structure and determinism in an efficient way. The average mutual information is used to determine the sampling rate. The embedding dimension is determined by singular system analysis.

In Chapter 3 nonlinear models are introduced. The main emphasis is on VS filters and on RBF networks. The topology of a VS filter and its training methods are described. The configurations of a normalised and an ordinary RBF network are shown. Different techniques for estimating the location and the bandwidths of the kernels are presented. A variety of simulations are carried out to determine a fast and reliable clustering procedure. Finally least squares techniques are discussed to train up the linear weights in the RBF network.

The investigation into Broomhead's filter method is presented in Chapter 4. The scenario is a narrowband signal in a communication channel corrupted by broadband noise. The noise is assumed to be nonlinear and

deterministic. Three different experiments are carried out and compared with two different linear filtering techniques. The restrictions and limitations of the filter method are shown. Finally conclusions are drawn from the results.

In Chapter 5 are variety of schemes to circumvent the restrictions of Broomhead's filter method are presented. The first one is the new filter method. It is explained in detail and simulation results for different input SNR and linear and nonlinear filters are shown. The next one uses cascaded RBF networks. This method is explained and simulation results are presented at the end. A linear filtering technique compares the performance of this filter method. Another filter method which makes use of a quantiser is presented next. The underlying idea and the design of the quantiser are shown. Simulation results are presented at the end. Finally a filter method for recovering and detecting signals of interest in heavy nonlinear deterministic noise is presented. The scheme of the filter method is depicted and simulation results are shown. A comparison with linear techniques is also presented.

Nonlinear active noise control is presented in Chapter 6. A brief introduction is given into ANC. A background section provides the state of the art of ANC in ducts. Conventional linear control techniques are reviewed and simulated. Alternative control strategies which allow a nonlinear controller to be implemented are investigated and simulated. A new control scheme is designed to allow an adaptive block least squares training method in conjunction with a nonlinear controller. A mathematical expression of a combined linear and nonlinear controller is derived and verified in simulations. Simulation results with a variety of different controllers and noise sources are presented. A discussion of the results and a summary end this chapter.

Chapter 7 concludes this thesis. It presents a review over the work carried out in this thesis and discusses the achieved results. Further, suggestions and comments are made on possible future work.

Chapter 2

Nonlinear Dynamics and Chaos

A dictionary definition of chaos is a 'disordered state of collection; a confused mixture'. This is an accurate description of dynamical systems theory today - or any other lively field of research.

Morris W. Hirsch

2.1 Introduction

The last three decades have seen a flood of publications on analysing [2], detecting [3], filtering [4], controlling [5], predicting [6] and modelling [7] chaos. Research into the *chaos phenomena* is carried out in almost every scientific discipline. Due to the advances in computer technology complex numerical methods are readily implemented [8,9]. Heart rate variability [10], number of sun spots [11], DC/DC buck converters [12], parameters in artificial neural networks [13], ocean 'ambient' noise [14], engine noise [15], electronic circuits [16], speech [17], sea clutter in radar signals [18] to name a few research areas have been under investigation to determine any chaotic behaviour. A few signals, *i.e.* speech [17] and heart rate variability [10] have been classified as nonchaotic signals, whereas for example sea clutter [18] or fan noise [19] have been shown to be chaotic processes. The analysis tools used to determine chaotic behaviour require stationarity in the measured data which is not always the case. Also, the measured signal is usually contaminated with stochastic noise, plus the lack of sufficient long data sets have often mislead researchers. A drawback with the analysis tools in nonlinear dynamics is that they do not provide any uncertainty limits, *i.e.* standard deviation. If the researcher is not careful enough in choosing the appropriate parameters for the analysis tools misleading results are possible.

Another popular aspect of chaos is in exploiting its properties. Its broadband nature is used in masking signal of interests (chaotic modulation schemes [20]) and its sensitivity to very small perturbations in control techniques [21]. The determinism [22] in chaos is exploited in fractal coding [23,24] and in signal separation [25]. Chaotic processes become unpredictable in the long term, but it is possible to predict a chaotic time series in the short term. This offers new strategies to model chaotic processes [6,26–32].

Chaotic processes arise from nonlinear deterministic systems. Nonlinear is defined as the negation of linear. Linearity in algebra, in terms of functions, is defined as follows : $f(x + y) = f(x) + f(y)$ and $f(ax) = af(x)$. Whereas in geometry, linearity is referred to Euclidean objects, *i.e.* lines and planes, and nonlinearity to more complex objects, *i.e.* spheres.

Determinism is another important and exploitable feature of chaotic processes. It is defined as a unique consequent for every state in a dynamical system. In other words, each observation is a function of the previous observations. This means the longer one observes a system the more one can learn from it. The ability to predict this kind of time series is the key difference to stochastic processes.

A dynamic system can be described by a set of differential equations [33–35]. The equations must be nonlinear to generate chaotic solutions, but apart from that can be remarkably simple [36]. For autonomous (not dependent on time) systems the ordinary differential equation (ODE) has to be at least of 3-D (3 degrees of freedom) for chaotic behaviour. Non-autonomous system of 2-D can exhibit chaos. As for discrete systems like the logistic map, one can think of them as the Poincare maps of 2-D flows, therefore only 1-D for difference equations is necessary to generate chaos.

One feature which characterises chaotic behaviour is that it is sensitive to initial conditions. Imagine a system to be started twice, but from slightly different initial conditions. For nonchaotic systems this uncertainty leads only to an error in prediction that grows linearly with time. For chaotic systems, on the other hand, the error grows exponentially in time, so that the space of the system is essentially unknown after a very short time.

Another feature of chaotic processes is that they exhibit broadband spectra and masquerade as random time series when analysed with linear techniques [37]. Consider for instance the two aperiodic time series in Figure 2.1(a) and (b). At first sight both time series appear to be random noise.

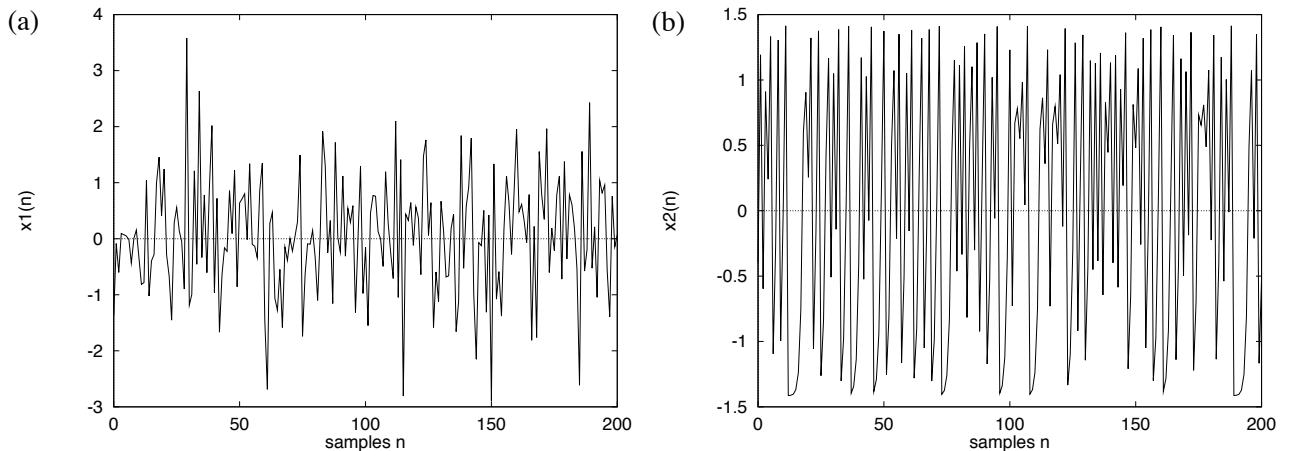


Figure 2.1: Aperiodic time series : (a) Time series 1 (b) Time series 2

The power spectral densities (PSD) of the two time series are shown in Figure 2.2(a) and (b), respectively. The two PSD plots show that the energy of both time series are equally distributed over the whole frequency range.

This is an indication of broadband random noise.

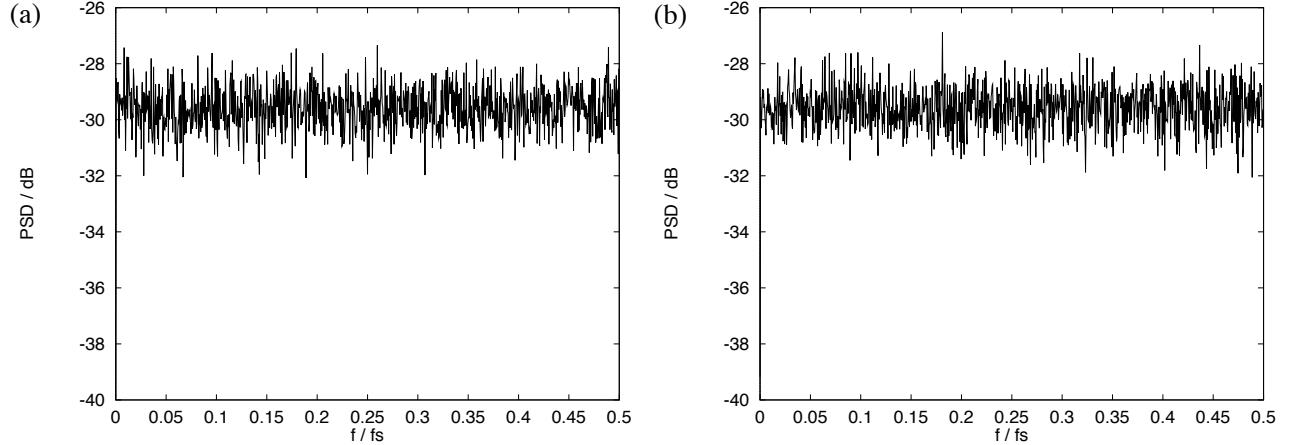


Figure 2.2: PSD (averaged 50 times) : (a) Time series 1 (b) Time series 2

The autocorrelations $R_{xx}(l)$ of both time series in Figure 2.3(a) and (b) have only a significant peak at lag zero, which implies that both time series are uncorrelated.

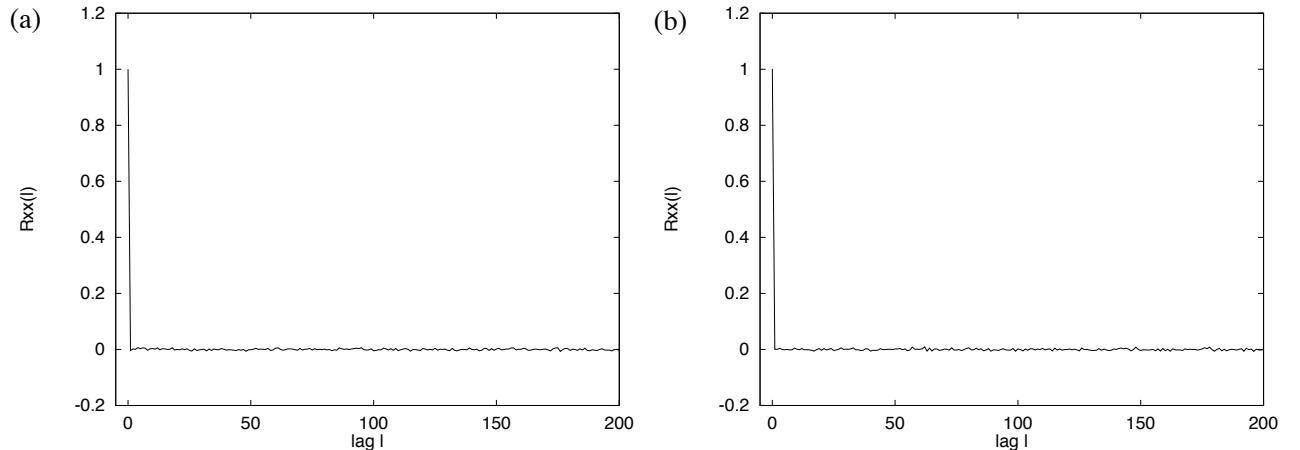


Figure 2.3: Autocorrelation (averaged 50 times) : (a) Time series 1 (b) Time series 2

After applying these 2nd order statistical tests to both time series, it would be quite natural to assume that both time series are purely stochastic processes. To gain further insight into the nature of the time series higher order statistics (HOS) [38–40] could be applied. Although HOS are able to detect nonlinearities [38,41,42], they are at times difficult to interpret [43,44]. This is due to high variance of the estimates of the HOS [45] often caused by the lack of large data sets. Therefore, HOS techniques tend not to be very practical. However, there are other analysis tools from the theory of nonlinear dynamics [9] which can be readily applied. One method is the method of delays which is described at the end of this chapter. This method is able to construct a phase plane of a nonlinear dynamic system. The phase planes are shown in Figure 2.4(a) and (b). It is clear that the phase plane in the Figure 2.4(b) has a low dimensional structure. This structure is called a strange attractor

which will be explained later on. In this case it is produced by the Logistic map normalised to have zero mean. This simple map, given by the difference equation

$$x(n+1) = \mu x(n)(1 - x(n)), \quad x(n) \in [0, 1] \quad (2.1)$$

arises from the corresponding differential equation

$$\frac{dx}{t} = \mu x(1 - x) \quad (2.2)$$

The Logistic map is nonlinear and depending on the value μ exhibits equilibrium points, period doubling, bifurcations and chaotic behaviour [33, 46–48]. To produce chaotic behaviour μ is chosen to be 4.0. Figure 2.4(a) is produced by white Gaussian noise and covers almost the whole state space.

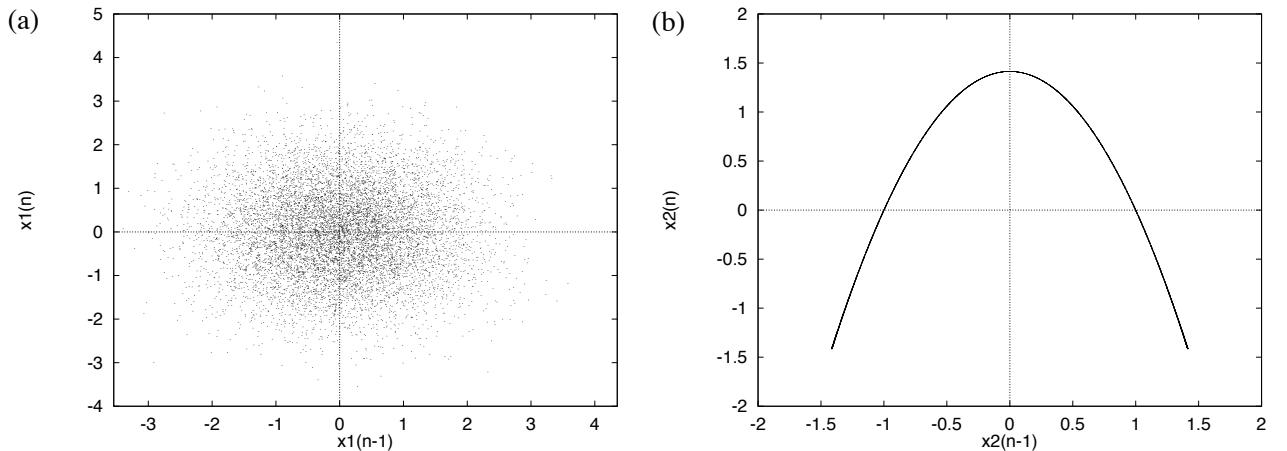


Figure 2.4: Phase plane : (a) Time series 1 (b) Time series 2

A useful tool for analysing the outcomes of nonlinear dynamic systems with varying parameters is the Feigenbaum diagram [33]. In Figure 2.5 the Feigenbaum diagram of the Logistic map is displayed. It shows the outcomes of the Logistic map in a range of $\mu = 2.4 \dots 4.0$ in steady state. In the range $\mu = 2.4 \dots \sim 3.0$ the steady state is an equilibrium point. At $\mu \simeq 3.0$ a bifurcation occurs and the outcomes oscillate between two steady states. Increasing μ at this point produces chaotic behaviour, interrupted by further period doublings.

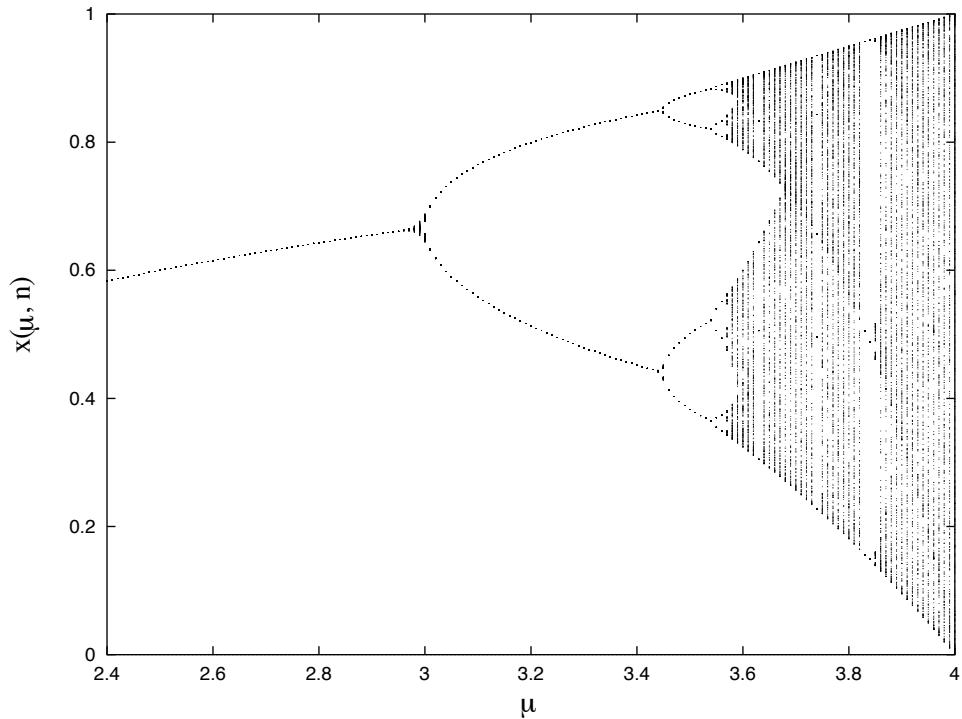
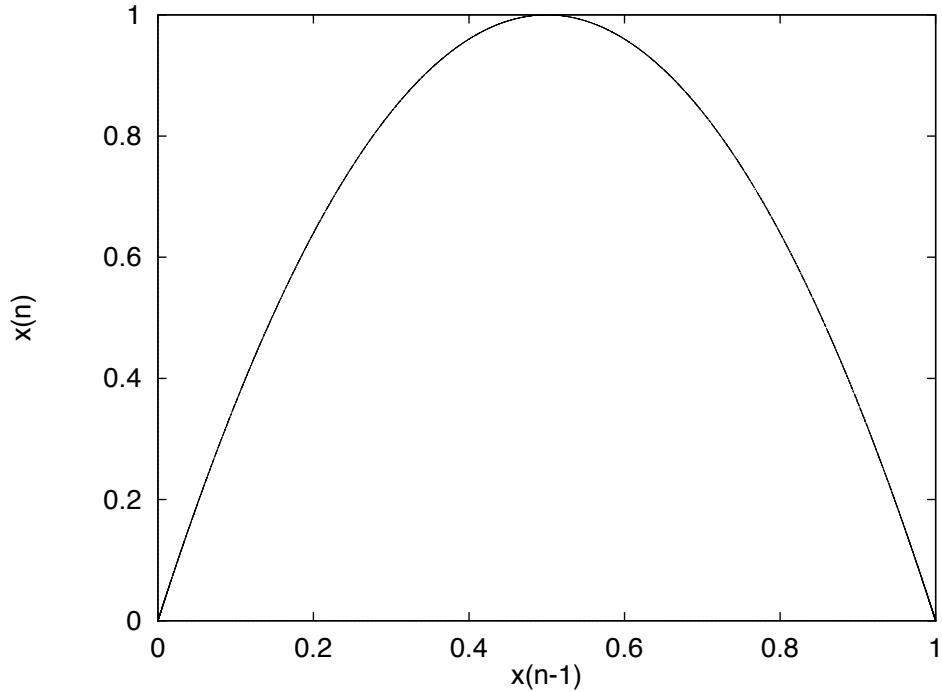


Figure 2.5: Feigenbaum diagram of the Logistic map

The attractor of the Logistic map is shown in Figure 2.6. The parameter μ is chosen to be 4.0, so that the Logistic map produces chaotic behaviour.

Figure 2.6: Attractor of the Logistic map ($\mu = 4.0$)

Another famous attractor is shown in Figure 2.7. It is the three-dimensional Lorenz attractor. It can be seen that the initial parameters $(x(0), y(0), z(0))$ lie in the basin of attraction. The trajectory is attracted to the Lorenz attractor and stays there for an infinite time.

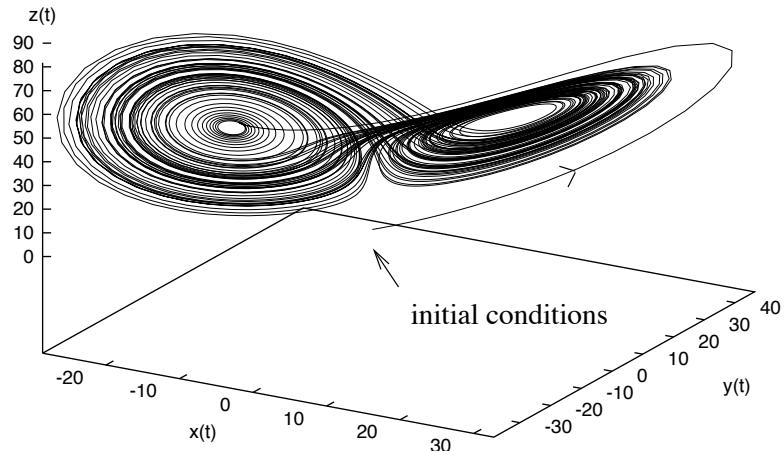


Figure 2.7: The Lorenz attractor

2.2 History

Henri Poincaré (1854-1912) was said to be the first one who carried out mathematical research in chaos. At the time around 1890 he was investigating the stability of the solar system. His main question was, if the planets would continue to stay on their orbits or would eventually drift off. To find an answer (which he did not find) he invented a new analytical method, the geometry of dynamics. This method formed the theoretical foundations of the subject of topology. Chaotic behaviour was discovered by Poincaré in the orbital motion of three bodies which mutually exert gravitational forces on each other. Other scientists followed his path. Andrey Kolmogorov, a mathematician, made basic advances in the irregular features of dynamics.

In the 1960's a plan to classify all kinds of typical dynamic behaviour was proposed by Stephen Smale. Edward Lorenz was the first one to publish a strange attractor in the *Journal of the Atmospheric Sciences*. Unfortunately, this journal is usually not read by mathematicians or physicists and, therefore, the research on chaos was delayed by a decade or so. Lorenz numerical work described a simplified model for convection. He studied its implications for weather prediction.

The concept of chaos in practical applications in the 1970's followed. For example, sudden irregular beha-

viour, for no apparent reason, in smooth flowing fluids where explained by David Ruelle and Floris Takens as chaotic behaviour. Other experimental scientists, notably Harry Swinney, Jerry Golub and Albert Libchaber followed the same path [49].

Up to the 1970's experiments, *e.g.* periodic chemical reactions, which produced irregular patterns of processes, hence chaos, were usually dismissed as an unsuccessful experiment. That has changed rather dramatically in the light of the chaos theory. The word 'chaos' was coined by James A. Yorke and Tien-Yien Li in 1975, whereas the phrase 'strange attractor' was created by D. Ruelle and F. Takens in 1971.

2.3 Invariants of the Dynamics

To classify different physical systems it is necessary to obtain invariant quantities [50]. Invariant quantities are independent from the set of initial conditions of a dynamical system. There are three different approaches for the purpose of defining invariant measures on the attractor :

- Density estimation
- Dimension
- Lyapunov Exponents

The invariant measure from density estimation is defined as the natural density of points in the state space. It describes where the orbit has been, and counts how many points within a volume in the state space there are. A detailed definition may be found in [51]. The more widely used invariant quantities dimension and Lyapunov exponents are described in more detail in the following sections.

2.3.1 Dimension

Attractor dimension [52] is an invariant quantity and a classifier for dynamical systems. The simplest concept of dimension is the number (integer) of coordinates needed to specify the state space. Chaotic attractors generally have a noninteger dimension as well as a complex geometry, and are therefore called strange attractors. The Cantor set in Figure 2.8 illustrates the concept of having non-integer dimensions. The procedure goes as follows. Remove every middle third of a line segment and repeat infinitely. In contrast to a line with its infinite number of points and finite length, the Cantor set has infinite number of points but zero length. Therefore its dimension has to be between 0 and 1 ($d_{\text{Cap}} = \log 2 / \log 3$).

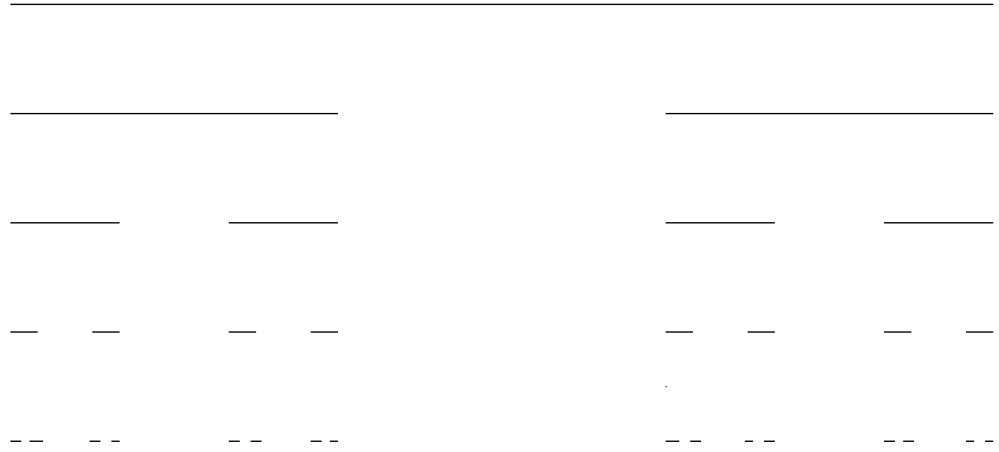


Figure 2.8: Cantor set

There are different definitions and ways to determine the fractal dimension of an attractor.

- *Capacity dimension* (box counting method)

This dimension measure is a basic one and is defined as follows. It depends on how many d dimensional hypercubes are needed to cover the attractor. For example a line of length l may be covered by N line segments with a length of ε .

$$N(\varepsilon) = \frac{l}{\varepsilon}$$

A square of side l in two dimensions requires $N(\varepsilon) = (l/\varepsilon)^2$ squares with width ε to be covered. For d dimensions this can be generalised to

$$N(\varepsilon) = \left(\frac{l}{\varepsilon}\right)^d$$

The capacity dimension is defined in the following way

$$d_{\text{Cap}} = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log(1/\varepsilon)}$$

This measure d_{Cap} , for an accurate result, requires the complete knowledge of the attractor. Measured or calculated attractors represent usually only temporal sections. It also is very difficult to compute the measure d_{Cap} for strange attractors [49]. Therefore, for a more sophisticated practical definition the two following dimensions include a probabilistic term.

- *Information dimension*

This definiton takes into account on how often a trajectory visits a particular volume element in state space. An utilised entropy measure is defined as follows

$$H(\varepsilon) = - \sum_{i=1}^{N(\varepsilon)} P_i \ln P_i$$

where P_i is the probability that a trajectory has visited the i^{th} volume element. $N(\varepsilon)$ are the volume elements with diameter ε . Thus, the information dimension is defined as

$$d_{\text{Inf}} = \lim_{\varepsilon \rightarrow 0} \frac{\log H(\varepsilon)}{\log(1/\varepsilon)}$$

- *Correlation dimension*

This measure is the most popular one due to its ease in computation. A well known implementation of the algorithm by Grassberger *et al.* may be found in [53,54]. A correlation function $C(\varepsilon)$ is given by

$$C(\varepsilon) = \lim_{N \rightarrow \infty} \left[\frac{1}{N^2} \sum_{i,j=1}^N \varphi(\varepsilon - \| \mathbf{s}_i - \mathbf{s}_j \|) \right] \quad (2.3)$$

where \mathbf{s}_i and \mathbf{s}_j are points on the attractor and $\varphi()$ is the Heavyside step function. The definition for the correlation dimension is given as

$$d_{\text{Cor}} = \lim_{\varepsilon \rightarrow 0} \frac{\log C(\varepsilon)}{\log \varepsilon}$$

- *Lyapunov dimension*

This measure was conjectured by Kaplan *et al.* after a series of numerical experiments [49]. It is defined as follows

$$d_{\text{Lyap}} = m + \frac{1}{|\lambda_{m+1}|} \sum_{k=1}^m \lambda_k$$

where m is the maximum integer such that $\sum_{k=1}^m \lambda_k \geq 0$ and λ_i are the Lyapunov exponents ordered by magnitude. Kaplan *et al.* claim that the Lyapunov dimension d_{Lyap} is generally equal to the Information dimension d_{Inf} .

2.3.2 Lyapunov Exponents

The Lyapunov exponent [55–57] of a dynamic system is a measurement quantity for the sensitivity dependence upon initial conditions that is characteristic of chaotic behaviour. In Figure 2.9 the first fifty samples of two time series generated by the Logistic map ($\mu = 4.0$) are shown. The difference between the initial conditions of the two time series is only 10^{-09} and it can be seen that the time series diverge after around 25 samples and run on completely different trajectories.

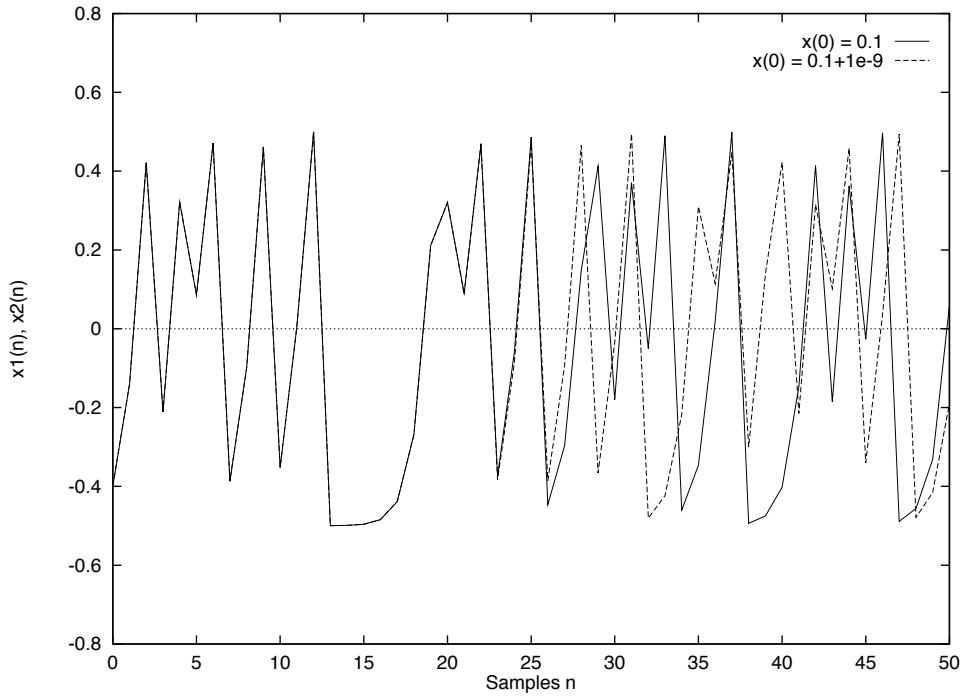


Figure 2.9: Sensitivity to initial conditions

The Lyapunov exponent λ may be readily computed for a one dimensional map such as the Logistic map. If a system is allowed to evolve from two slightly differing initial states, x and $x + \epsilon$, then after n iterations their divergence may be characterized approximately as

$$\epsilon(n) \approx \epsilon e^{\lambda n} \quad (2.4)$$

where the Lyapunov exponent λ gives the average rate of divergence. If λ is negative, slightly separated trajectories converge and the evolution is not chaotic. If λ is positive, nearby trajectories diverge; the evolution is sensitive to initial conditions and therefore chaotic. A one dimensional map is given as

$$x(n+1) = f(x(n)) \quad (2.5)$$

The difference between two initially nearby states after n th steps is written as

$$f^n(x + \epsilon) - f^n(x) \approx \epsilon e^{n\lambda} \quad (2.6)$$

$$\ln \left[\frac{f^n(x + \epsilon) - f^n(x)}{\epsilon} \right] \approx n\lambda \quad (2.7)$$

For small ϵ , this expression becomes

$$\lambda \approx \frac{1}{n} \ln \left| \frac{df^n}{dx} \right| \quad (2.8)$$

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \quad (2.9)$$

The Lyapunov exponent gives the stretching rate per iteration, averaged over the trajectory. For a n -dimensional maps there are n Lyapunov exponents, since stretching can occur for each axis. The sum of Lyapunov exponents for a dissipative system must be negative. If the system is chaotic then at least one of the exponents has to be positive. The largest Lyapunov exponent gives an indication of how far into the future predictions can be made.

2.4 Embedding Dimension

Usually in practice there is only a measured time series from a physical system available [58,59]. The underlying dynamics (differential equations) are unknown to the observer. Fortunately, it is possible to extract invariant quantities from a measured time series. This is based on Taken's delay embedding theorem which allows to reconstruct the attractor of a dynamical system. The following section gives a brief review on Taken's delay embedding theorem. Section 2.4.2 explains a technique to determine the embedding dimension and how this technique can also be used to project the data onto a noise-reduced sub-dimensional space. Another technique, false nearest neighbours, to determine the embedding dimension is briefly reviewed in Section 2.4.3.

2.4.1 Taken's Delay Embedding Theorem

One of the reasons why the chaos phenomenon has been studied not only by physicists and mathematicians is because of the delay embedding theorem from Takens [60]. It provides the theoretical foundations to obtain quantitative dynamics from experimental data in a rather simple way. The basic idea is to construct the state space of a dynamical system by time series observations. If the attractor of a dynamical system is contained within a finite dimensional manifold M , with $\dim M = m < \infty$, then an embedding of M can be constructed from a time series using a delay register. In order to explain an embedding [61] it is necessary to give a brief review of definitions from differential topology.

- **Invertible map** A map $f : X \rightarrow Y$ is *one to one* if no two points in X map to the same point in Y . The map $f : X \rightarrow Y$ is *onto* if for every point $y \in Y$ there exists at least one point in X that is mapped to y by f .
A map that is one to one and onto is called *invertible*. If f is invertible, the inverse map, $f^{-1} : Y \rightarrow X$, is also one to one and onto.
- **Smooth map** Let $U \subset R^m$ be an open set. A map $f : U \rightarrow R^p$ is *smooth* if it possesses continuous partial derivatives of all orders.

- **Homomorphism** Linear map between vector spaces $L : X \rightarrow Y$. L is a *homomorphism* because it preserves structural properties, like vector addition $L(x_1 + x_2) = Lx_1 + Lx_2$
- **Diffeomorphism** A smooth map $\phi : X \rightarrow Y$ is a *diffeomorphism* if it is invertible and if the inverse map $\phi^{-1} : Y \rightarrow X$ is also smooth.
- **Manifold** A *k-dimensional manifold* M is a set of points that locally resembles R^k . More precisely, M is a k-dimensional manifold if for each point $x \in M$, there exists an open neighbourhood U of x such that U is diffeomorphic to some open neighbourhood in R^k .

Embedding A set $U \subset R^k$ is *compact* if it is closed and bounded. Examples of compact sets are the circle and the sphere. For $f : X \rightarrow Y$, the *pre-image* of a set $Z \subset Y$, denoted by $f^{-1}(Z)$, is the set of all points in X that f maps into Z , that is,

$$f^{-1}(Z) = \{x \in X : f(x) \in Z\}. \quad (2.10)$$

A map is *proper* if the pre-image of every compact set in Y is compact.

Let X and Y be two manifolds with $\dim(X) < \dim(Y)$. A map $f : X \rightarrow Y$ is an *immersion* if its derivative $Df(x)$ is of full rank for every $x \in X$.

Let X be a compact manifold. If $f : X \rightarrow Y$ is an immersion, proper, and one to one, then it is called an *embedding*. An embedding $f : X \rightarrow Y$ maps X diffeomorphically onto a manifold $f : X' \subset Y$.

The embedding described by Takens is a map $\zeta_{2m+1} : M \rightarrow R^{2m+1}$, defined by

$$\zeta_{2m+1}(x_0) = (v(x_0), v(x_{-1}), \dots, v(x_{-2m})) \quad (2.11)$$

where $v(x_0), v(x_{-1}), \dots, v(x_{-2m})$ are a sequence of real-valued measurements $v : M \rightarrow R$ and $\dots, x_{-1}, x_0, x_1, \dots$ is a trajectory on the dynamical system on M .

2.4.2 Singular System Analysis

This technique is also known under the names principal component analysis (PCA), Karhunen-Loëve decomposition and principal value decomposition [62, 63]. The algorithm is a projection of the delay coordinates. It decomposes multidimensional data into linearly independent coordinates. Therefore it eliminates linearly dependent coordinates and artificial symmetries [2]. The set of eigenvalues, called the singular spectrum, are derived from the following covariance matrix

$$COV = \frac{1}{N} \sum_{n=1}^N (x(n) - \bar{x})(x(n) - \bar{x})^T \quad (2.12)$$

with

$$\bar{x} = 1/N \sum_{n=1}^N x(n) \quad (2.13)$$

where $x(n)$ is the time series to be analysed.

This technique can also be carried out locally, where the covariance matrix is over a neighbourhood of N_B nearest neighbours [51]. In some cases singular system analysis can reduce noise. In [62] it is shown that, if the noise is a Gaussian i.i.d. noise process the singular system analysis acts as an optimum linear coordinate transformation.

In order to chose an appropriate embedding dimension it is necessary to determine the degrees of freedom in the time series. Usually a measured time series is contaminated with stochastic noise. The high dimensional stochastic noise will fill more or less uniformly any low-dimensional space and, therefore, it will obscure any low dimensional chaotic process. The singular system analysis is also able to detect noise-dominated coordinates. Broomhead *et al.* in [2] discovered that the singular spectrum decreases until it hits a plateau. The eigenvalues in this plateau are roughly equal and are thought to be the outcomes of having Gaussian noise in the data. The results in Figure 2.10 show the eigenvalues of the Lorenz chaos with and without Gaussian noise. It is clear in both cases that the chaos is low-dimensional and the estimated embedding dimension is three.

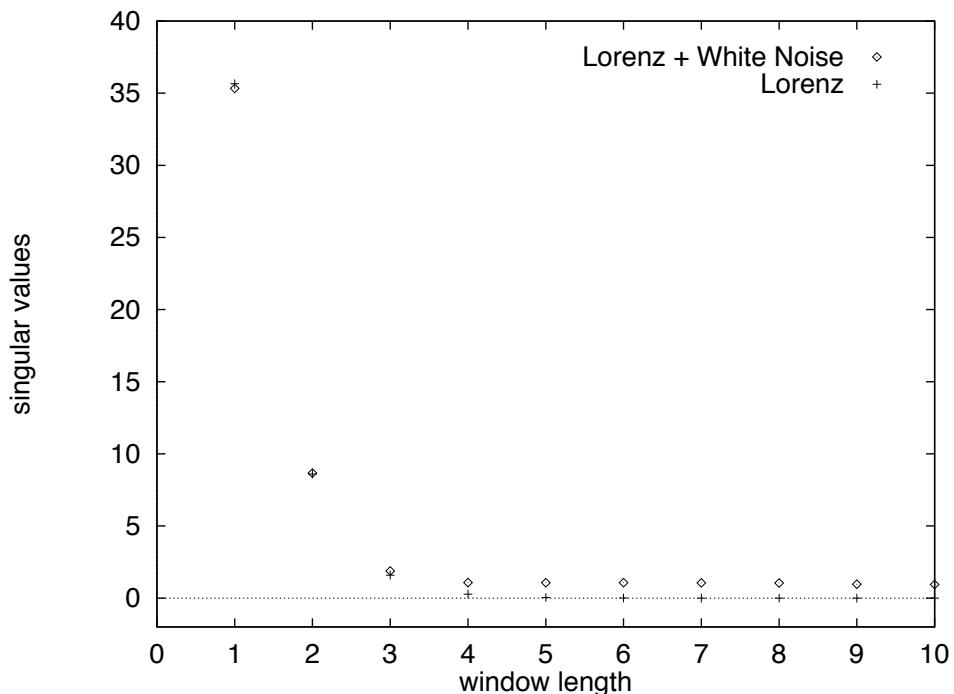


Figure 2.10: Singular System Analysis

Although the singular system analysis provides the correct embedding dimension in this scenario, it is only able to give a "linear hint" towards the numbers of active degrees of freedom in a time series. It is also sometimes difficult to interpret [9]. Therefore, one should be careful when using this technique to determine the embedding dimension.

2.4.3 False Nearest Neighbours

This method of determining the embedding dimension d_{Emb} comes directly from the data when projected onto a low dimensional space. The main question is : Are all the false crossings of the trajectories with itself eliminated which were due to the fact that the chosen embedding dimension was too low. The behaviour of near neighbours (points on the trajectories) under changes in the embedding dimension from $d \rightarrow d + 1$ is monitored by this method. When the number of false nearest neighbours drops to zero, the attractor is unfolded or embedded in R^{Emb} , a d_{Emb} Euclidean space. A detailed explanation and implementation of the procedure may be found in [64, 65].

2.5 Sampling Rate

The embedding theorem states that any sampling rate will be acceptable is not very useful for the purpose in extracting invariant quantities from a measured time series. If the time lag T between two adjacent samples is too small, then the coordinates $x(n)$ and $x(n+T)$ will be so close to each other in numerical value so that it will be difficult to distinguish them. On the other hand, if the time lag T is too large, then $x(n)$ and $x(n+T)$ will be completely independent and, therefore, the projected orbit will be spanned onto two unrelated directions. A first hint to determine the lag T is to use the autocorrelation function

$$C_L(\tau) = \frac{1/N \sum_{m=1}^N [x(m+\tau) - \bar{x}][x(m) - \bar{x}]}{1/N \sum_{m=1}^N [x(m) - \bar{x}]^2} \quad (2.14)$$

where

$$\bar{x} = 1/N \sum_{m=1}^N x(m) \quad (2.15)$$

An appropriate value for the lag T would be the first zero crossing of the autocorrelation function $C_L(\tau)$. This procedure gives a linear hint of the independance of the coordinates. The time series which posses chaotic behaviour are generally nonlinear. For this reason a nonlinear notion in independence would be more appropriate. Such a kind of measure is derived from Shannon's information theory and is described in the following section.

2.5.1 Average Mutual Information

The sampling rate of a chaotic time series is determined in a nonlinear way by the average mutual information. It measures the independence between two samples. The average mutual information is a kind of generalisation to the nonlinear world from the correlation function in the linear world. It has been suggested [66] that the optimum sampling rate is at the first minimum of the average mutual information $I(T)$.

Taking a measurement a_i , the amount of information in bits about another measurement b_k is given by

$$I_{AB}(a_i, b_k) = \log_2 \left[\frac{P_{AB}(a_i, b_k)}{P_A(a_i)P_B(b_k)} \right] \quad (2.16)$$

where $P_A(a)$ is the probability of observing a out of the set A and $P_B(b)$ the probability of finding b in a measurement of B . The joint probability of the measurement a and b is $P_{AB}(a, b)$. $I_{AB}(a_i, b_k)$ is defined as the mutual information between two measurements. This abstract definition may be used for two samples in a time series as follows.

The average amount of information about $x(n+T)$ when making the observation $x(n)$ is

$$I(T) = \sum_{n=1}^N P(x(n), x(n+T)) \log_2 \left[\frac{P(x(n), x(n+T))}{P(x(n))P(x(n+T))} \right] \quad (2.17)$$

and $I(T) \geq 0$. When the observations $x(n)$ and $x(n+T)$ are completely independent then $I(T) = 0$. Figure 2.11 shows the average mutual information of the Lorenz data. The first minimum occurs at $T \approx 0.15$. The Lorenz data is sampled with 100Hz. This means that every 15th sample is sufficient to gain independent (not statistically) coordinates.

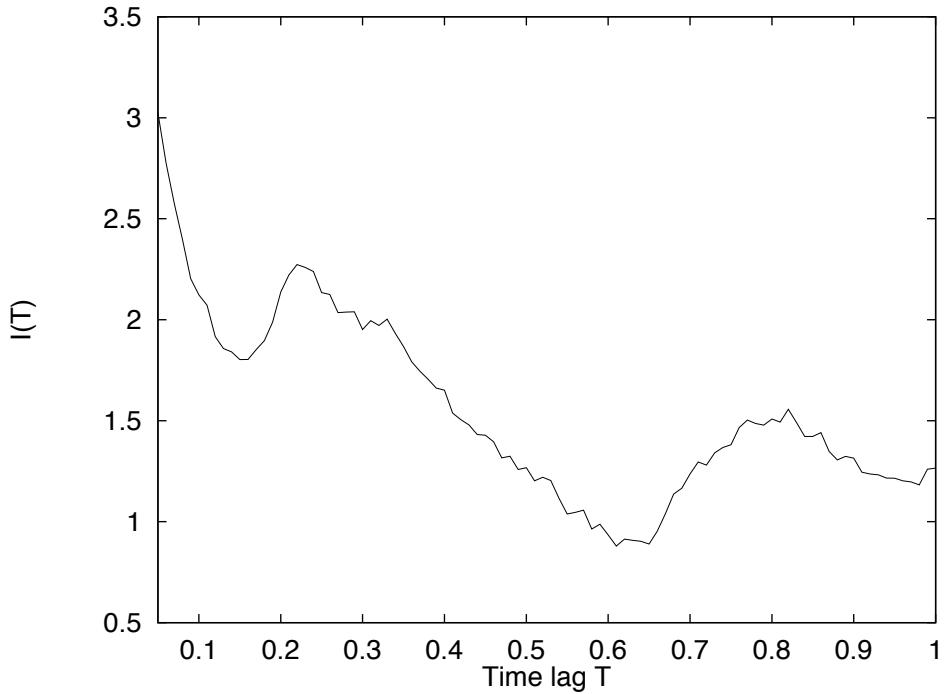


Figure 2.11: Average Mutual Information $I(T)$ for Lorenz data

This choice of lag T is only "prescriptive" and cannot be generalised. For certain dynamical system there are no minimas in the average mutual information $I(T)$ and, therefore, other criteria have to be used. Another point is the choice of lag for the second, third and so on delay. The delays in this thesis are chosen to be equal and are determined by the first minimum of $I(T)$ in the following chapters.

2.6 Summary

This chapter has given a brief overview of the field of nonlinear dynamics with a focus on chaos. The 'young' history of the research on chaos has been outlined. Invariant quantities have been described and techniques to determine them. A brief review of Taken's delay embedding theorem has been given, together with two different techniques, singular system analysis and false nearest neighbours, to estimate the embedding dimension. A general method, the average mutual information, to determine an optimum sampling rate was presented.

Chaos theory is still evolving and at the moment is not yet well defined. However, it explains irregular patterns in processes which a few decades ago were discarded as some error in an experiment. Noisy signals in signal processing are often encountered. Noise or interference does not contain any information and contaminate other signals of interest. The noise or interference is usually assumed to be a white Gaussian process. However, the disturbance with its irregular behaviour may well be a chaotic process and, hence, other nonlinear filtering or control techniques may be more suitable.

Chapter 3

Nonlinear Models

3.1 Introduction

Linear filters have played a very important role in signal processing. In many scenarios, *e.g.* filtering a signal corrupted with additive white Gaussian noise, equalisers for minimum phase channels, etc., linear models provide the optimum model in the least square sense. The obvious advantage of linear filters is their inherent simplicity. In general the computational complexity of linear models is much lower than that of nonlinear models. Design, analysis, and implementation of linear filters are relatively straightforward tasks in many applications.

However, there are several situations in which the performance of linear filters is unacceptable. Nonlinear processes are widespread in nature. Nonlinear models which, for example, predict nonlinear time series [67–70], identify nonlinear systems [71, 72], equalise high speed communication channels [73] or model a nonlinear function [7, 74] are clearly more suited than linear models. A vast amount of nonlinear applications using artificial neural nets [75–77], fuzzy logic [13, 78] and genetic algorithms [79] have occurred during the last 2 decades. Nonlinear models are usually very complex and are trained by nonlinear optimisation techniques. These optimisation techniques are computationally expensive and often only perform sub optimally [80]. A different kind of nonlinear model is the one which is *linear in their parameters*. This means that these models can be trained by sophisticated least squares techniques. Two different types have been intensively used in the following simulations : The Volterra series (VS) filter [81, 82] and the radial basis function (RBF) network [83–87]. The VS filter and the RBF network can in theory [88] model a broad range of arbitrary nonlinear functions.

The nonlinear modelling and forecasting of time series data have a relatively recent history. The statistics community has constructed stochastic nonlinear models since about 1980. Independently, the dynamical systems community, motivated by the phenomenon of chaos, has constructed deterministic nonlinear models since 1987 [50].

Section 3.2 presents the VS filter and its implementation. The RBF network and a variety of training methods are introduced in Section 3.3. Section 3.4 summarises this chapter.

3.2 Volterra Series Filter

A VS filter is a polynomial model of nonlinearity [89–91]. This filter is very popular in adaptive nonlinear filtering, because of its simplicity [71, 72, 81, 82]. Apart from the topology of the VS filter, the only unknown parameters are the linear weights (Volterra kernels). The VS filter has a structure similar to a FIR filter. The only difference is that the input vector \mathbf{x}_N is additionally nonlinearly expanded by higher order (*e.g.* quadratic, cubic, etc.) terms.

The VS filter is a feedforward filter, by definition, whereas adaptive nonlinear filtering with feedback is still in an early stage of development. To characterise a nonlinear filter by the system's unit impulse response is impossible and therefore to design a stable system with feedback is very difficult.

For an input $x(n)$ and output $y(n)$ of a discrete-time and causal nonlinear system a VS expansion can be written as

$$\begin{aligned} y(n) = & h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) \\ & + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} h_2(m_1, m_2)x(n-m_1)x(n-m_2) + \dots \\ & + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \dots \sum_{m_p=0}^{\infty} h_p(m_1, m_2, \dots, m_p)x(n-m_1)x(n-m_2)\dots x(n-m_p) \\ & + \dots \end{aligned} \quad (3.1)$$

The $h_p(m_1, m_2, \dots, m_p)$ in equation (3.1), are called the p-th order Volterra kernels. Without loss of generality it may be assumed that the Volterra kernels are symmetric, *e.g.* $h_2(m_1, m_2) = h_2(m_2, m_1)$. In practise only truncated Volterra series expansions are used, so that (3.2) may be written as

$$\begin{aligned} y(n) = & \sum_{m_1=0}^{N-1} h_1(m_1)x(n-m_1) \\ & + \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} h_2(m_1, m_2)x(n-m_1)x(n-m_2) + \dots \\ & + \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \dots \sum_{m_p=0}^{N-1} h_p(m_1, m_2, \dots, m_p)x(n-m_1)x(n-m_2)\dots x(n-m_p) \end{aligned} \quad (3.2)$$

The kernel h_0 represents a DC-offset and is usually set to zero. A disadvantage is that the number of kernels is $O(N^P)$. The complexity of the filter grows rapidly with large values of N and P . The number of kernels can be calculated as follows

$$\begin{aligned} n_0 &= 1 \\ n_j &= n_{j-1} \left(\frac{N+j-2}{j} \right) \quad j = 1, 2, \dots, P \\ \text{number of kernels} &= \sum_{j=0}^P n_j \end{aligned} \quad (3.3)$$

A VS filter of order $P = 2$ and degree $N = 3$, hence with 2 delay elements, is shown in Figure 3.1.

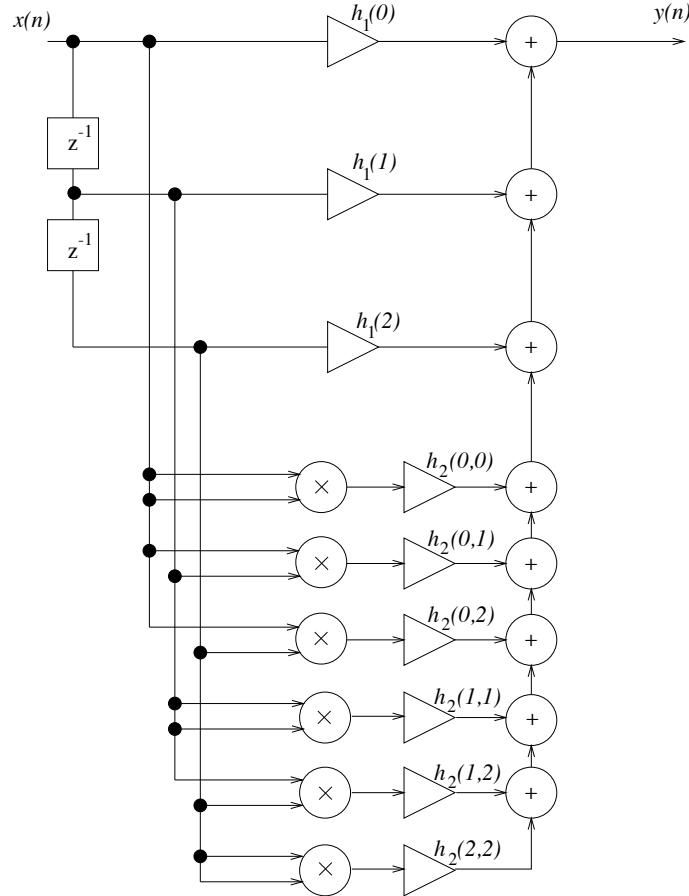


Figure 3.1: Second order Volterra filter with two delay elements

As mentioned earlier the VS filter (and the RBF network) is *linear in its parameters*. To demonstrate this feature (3.2) is rewritten as a vector inner product.

$$y(n) = \mathbf{h}^T \mathbf{x}_{\text{exp}}(n) \quad (3.4)$$

where

$$\begin{aligned} \mathbf{x}_{\text{exp}}^T(n) &= [x(n), x(n-1), \dots, x(n-N+1), \dots, x(n)x(n) \dots x(n), \dots, x(n-N+1)x(n-N+1) \dots \\ &\quad x(n-N+1)] \\ \mathbf{h}^T &= [h_1(0), h_1(1), \dots, h_1(N-1), \dots, h_p(0, 0, \dots, 0), \dots, h(N-1, N-1, \dots, N-1)] \end{aligned}$$

For this *linear in the parameter* nonlinear architecture, the weight vector \mathbf{h} can be estimated using a least squares algorithm.

To make the VS filter adaptive is straightforward, because adaptive least squares algorithms from linear adaptive models can be used. Two well known adaptive algorithms are the LMS and the RLS algorithm.

The LMS algorithm has low computational complexity and slow convergence in the least square sense and the RLS algorithm has high computational complexity and fast convergence.

The LMS and RLS algorithms for a second-order VS filter may be found in [89]. There, it is pointed out that the nonlinear expansion in the input vector \mathbf{x}_N will cause the eigenvalue spread to increase, even with a white input signal. Therefore, it is important to use algorithms whose convergence speed is independent of or less dependent on the statistics of the input signal. It is well known that the LMS algorithm suffers in convergence speed when the eigenvalue spread of the autocorrelation matrix is large. One approach to circumvent this dilemma is to use the RLS algorithm at the expense of high computational complexity.

The VS filter will fail if it has to model discontinuities, for instance a saturation type of nonlinearity. This is because the Volterra series expansion is a Taylor series expansion with memory which only fits the data well when the functions are smooth, in the sense that they are at least once differentiable (C^1) [89, 90].

3.3 Radial Basis Function Network

The radial basis function (RBF) network, like the multilayer perceptron (MLP) [92], is a static network [75, 88]. Static networks are memoryless which means that they have no feedback path. The MLP architecture is the most popular one in practical applications. However, it is more difficult to train than the RBF network, because the MLP network uses nonlinear optimisation techniques. This means that the parameters of the MLP network are often trapped in a local minima of the error function when using a gradient descent technique.

A RBF network consists of three layers. The input layer is made up of source nodes, which may be formed by a tapped delay line representing an embedding vector of an observed time series. The hidden layer nonlinearly expands the input signal. This nonlinear expansion is motivated by the fact, that a difficult nonlinear filtering problem may be nonlinearly transformed into an easier linear filtering one [88]. The output layer supplies the response of the network. The transformation from the input layer to the hidden layer is nonlinear, whereas the transformation from the hidden layer to the output layer is linear. If the kernels of the RBF network are fixed, then this kind of structure is sometimes referred to as *linear in parameters*. A block diagram of a RBF network is shown in Figure 3.2.

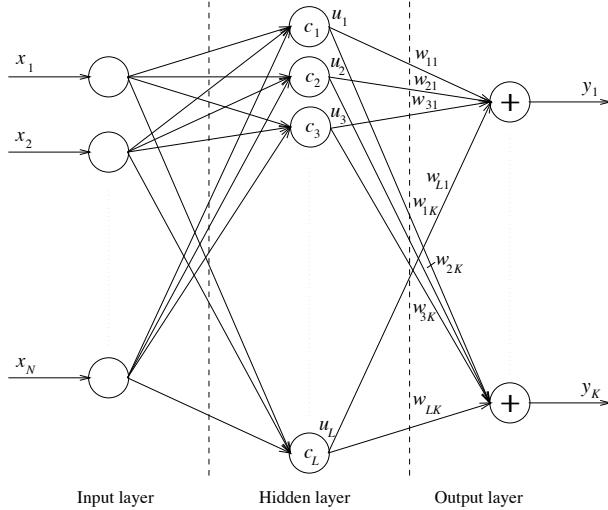


Figure 3.2: The radial basis function network

A RBF network for M inputs, one output and L kernels is defined as follows.

$$f(\mathbf{x}_M(n)) = \sum_{j=1}^L \mathbf{w}_j \mathbf{u}_j(n) \quad (3.5)$$

The most common nonlinear function for the hidden layer is a Gaussian kernel function of the form :

$$\mathbf{u}_j(n) = \exp -\frac{\|\mathbf{x}_M(n) - \mathbf{c}_j\|^2}{\sigma_j^2} \quad j = 1, 2, \dots, L \quad (3.6)$$

where \mathbf{u}_j is the output of the hidden layer, \mathbf{x} is the input vector and \mathbf{c}_j is the weight vector (centre of the Gaussian kernel). σ_j^2 is the scaling factor for the width of the Gaussian kernel. The type of nonlinearity is not crucial for the performance of the RBF network [93, 94].

In theory, the RBF network is capable of performing any kind of nonlinear transformation. It has been successfully implemented in classification [95–97], nonlinear dynamic modelling [98] and time series prediction [99] problems.

There are two different types of Gaussian RBF networks. The un-normalised Gaussian RBF network uses the exponential activation function, *i.e.* (3.6), to produce a localised *bump* as a function of an input vector $\mathbf{x}(n)$. In Figure 3.3(a) 4 Gaussian *bumps* with a small bandwidth σ^2 and in (b) with a large bandwidth σ^2 are shown. The other type is the normalised radial basis function (NRBF) with Gaussian kernels. The NRBF network uses the softmax [100] activation function, therefore, the outputs of all activation functions sum up to one. It is possible to interpret the outputs as posterior probabilities. The NRBF network features either a localised response like an un-normalised RBF network or non-localised behaviour like that of a sigmoid in a MLP network, depending on the location of the kernel. This behaviour leads to an enhanced approximation capability [101]. The NRBF network for M inputs and one output may be mathematically described as follows.

$$f(\mathbf{x}_M(n)) = \sum_{i=1}^L \mathbf{w}_i \Phi_i(n) \quad (3.7)$$

$$\Phi_i(n) = \frac{\exp(-\|\mathbf{x}_M(n) - \mathbf{c}_i\|^2 / \sigma_i^2)}{\sum_{j=1}^L \exp(-\|\mathbf{x}_M(n) - \mathbf{c}_j\|^2 / \sigma_j^2)} \quad (3.8)$$

In Figure 3.4(a) 4 normalised Gaussian kernels with a small bandwidth σ^2 and in (b) with a large bandwidth σ^2 are shown. The bandwidths of the kernels should be set to allow a smooth fit between the input space and the desired network outputs. Therefore, the bandwidth should be large enough so that neighbouring kernels overlap with their tails, but small enough to keep the response of a kernel localised. The bandwidth in Figure 3.3(a) is clearly too small, because there are regions in the input space which do not activate any of the kernels. On the other hand, the bandwidth in Figure 3.3(b) enables every kernel to be activated over the whole input space. This may cause problems when fitting a more complex target structure.

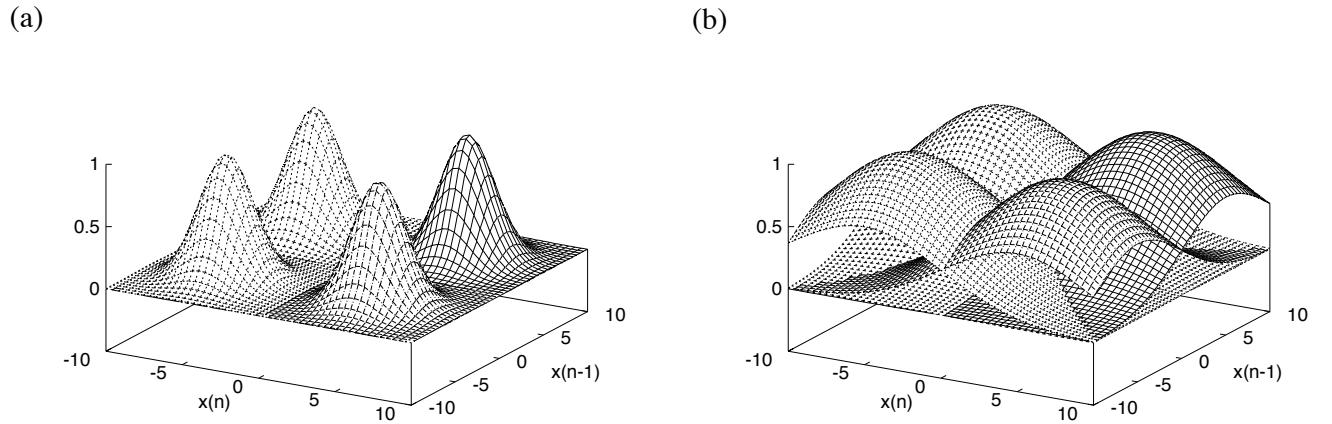


Figure 3.3: 4 un-normalised Gaussian basis functions with 2 inputs: (a) $\sigma^2 = 5$ (b) $\sigma^2 = 50$

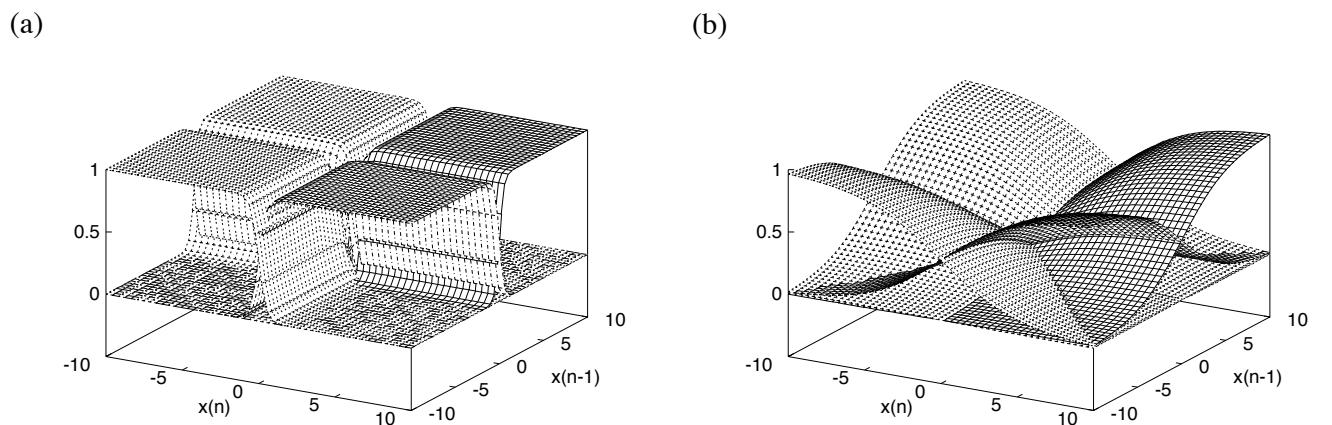


Figure 3.4: 4 normalised Gaussian basis functions with 2 inputs : (a) $\sigma^2 = 5$ (b) $\sigma^2 = 50$

The size (number of basis functions) of the RBF network is discussed in Section 3.3.2. Unsupervised training methods for the basis functions are introduced in Section 3.3.3. In Section 3.3.4 supervised training methods for the linear weights are discussed. Section 3.3.5 summarises this chapter.

3.3.1 Toplogy

This section describes procedures to combat the *overfitting* problem. *Overfitting* occurs when the nonlinear model is too complex and fits spurious quirks (*i.e.* *noise*) in the data. In other words, the nonlinear model does not provide a smooth mapping between the input signal and the desired signal. This means that the model will perform well with the training data, but may fail badly with some unseen data. The goal, therefore, is to select a model which is not too complex and, hence, will fit some spurious noise on the data, but be able to fully detect the signal in a complicated target set. Two methods from a variety [92, 102] are described.

The first method is called *early stopping* [103] and is a regularisation method. It can be summarised for a RBF network as follows :

- Divide the available data into three sets : training set, testing set and validation set
- Use a large number of Gaussian kernels
- Compute the testing error periodically during training
- Stop training when the testing error increases
- Estimate the generalisation error of the network with the validation set

The procedure is illustrated in Figure 3.5. A large RBF network is chosen, *e.g.* to model or predict a time series. The training error decreases as more training data is available. After each sample the linear weights are updated and then run with the testing set. The testing error will also decrease at first, but at a certain point (in Figure 3.5 at N samples) it will increase because the network is *overfitting* the training data. At that point the linear weights are stored and then validated with the third set, the so-called validation set.

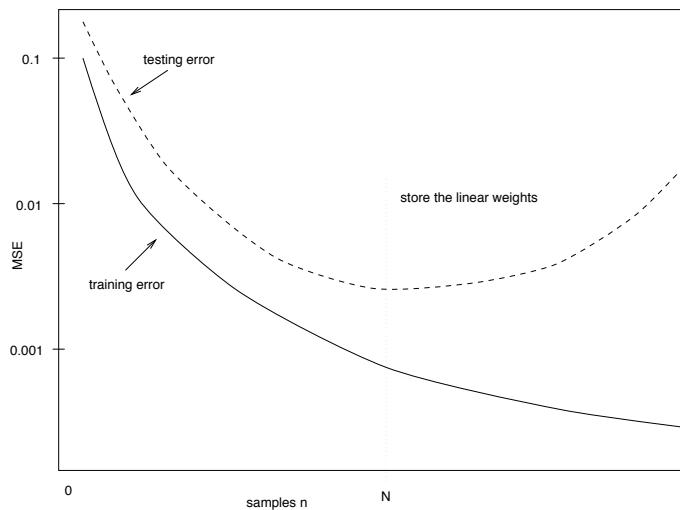


Figure 3.5: Early stopping

This method is fast and can be applied to networks in which the number of kernels exceeds the number of data points. Although this method seems to be rather *ad hoc*, it is closely related to the well-known statistical method of ridge regression [104].

Early stopping has been used in Chapter 4 and in various other simulations in Chapters 5 and 6.

The other method is a subset selection method, called orthogonal least squares (OLS). It is a supervised training method and was introduced by Chen *et al.* in [94] as a forward stepwise regression technique for selecting parsimonious RBF networks. The forward selection begins with zero kernels in the RBF network. At each step the kernel is added which most decreases the error function. It also is possible to start off with a large RBF network and then let the OLS algorithm determine which kernels may be discarded. An iterative OLS algorithm can be found in [94, 105]. In [106] the efficiency of the OLS algorithm for finding the smallest RBF network is investigated. The authors conclude that the OLS algorithm will not necessarily find the optimum configuration of kernels for a RBF network.

A brief experiment is carried out to compare the generalisation performance of the two techniques. The task is to predict one step ahead in the time series generated from the Logistic map (see Chapter 2). White Gaussian noise is added to the time series so that the resulting SNR is 20 dB. It should be noted that the additive white Gaussian noise is not predictable and, hence, will not be fitted by the NRBF network. The embedding dimension is 2 and the input space is covered with 203 kernels each with a bandwidth of $\sigma^2 = 0.28$.

In Figure 3.6 the *early stopping* method is shown. After taking in an initial block of data the training and the testing error are plotted for each sample. The linear weights are computed by a block least squares technique which is described in Appendix A. The linear weights which produce the best NMSE with the testing set are stored and validated by a third set from the input data. The validation NMSE is a single value and plotted on the y-axis. This example shows that the chosen RBF network with 203 kernels does not overfit the training data, no matter how many training samples are used. After around 800 samples the training error stops to decrease and, therefore, the testing error does not increase. The validation error is similar to the testing and training error and, therefore it can be concluded that the RBF network is a good generalised model for this specific prediction task.

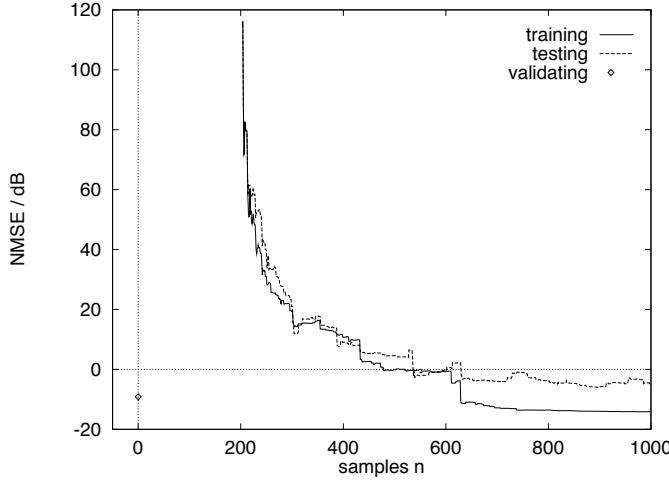


Figure 3.6: Early stopping : NMSE in dB for training, testing and validating

In Figure 3.7 the performance of the OLS algorithm is shown. As before, the same RBF network and prediction task are used. The method starts off with all 203 kernels and lets the OLS algorithm determine which kernels can be discarded without affecting the training error as less as possible. The NMSE in dB of the training and testing data for each discarded kernel, selected by the OLS algorithm is depicted. The linear weights for the best NMSE for the testing data is stored and then validated by a third set of the input data.

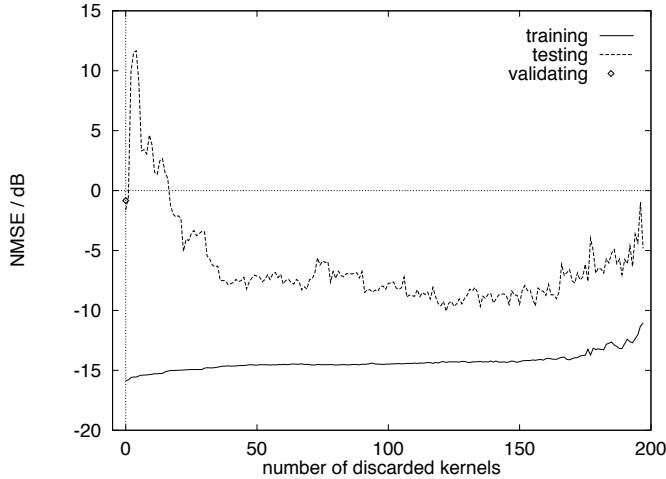


Figure 3.7: OLS : NMSE in dB for training, testing and validating

Although the OLS algorithm is able to select a parsimonious model which performs still very good with only 30 kernels on the training data it fails on the validation set. It is obvious from this experiment that the OLS algorithm produces a NRBF network which is not as robust as the *early stopping* method. This generalisation problem with subset selection methods has also been encountered in linear models subset selection methods which also do not generalise as well as regularisation methods [107].

3.3.2 Estimation of the Basis Function Parameters

The most common technique to train the kernels in a RBF network is the k-means algorithm [108, 109]. The algorithm partitions the input space into L clusters. The centres of the kernels are positioned onto the centres of those clusters. This is an unsupervised learning method based only on the input data. The traditional k-means clustering algorithm provides only a local optimum which depends on the initial positions of the kernels. Kernels may get trapped in a region of the input space where there are only a few data points. Those data points do not need a kernel and therefore the RBF network becomes too large and may be numerically ill-conditioned [110]. An adaptive k-means algorithm has recently been proposed [111] which converges to an optimal configuration independent of the initial positions of the kernels. This algorithm which incorporates a dynamic adjustment of the learning rate may be summarised as follows.

- Initialization

$$\sigma_j^2 = \text{distance to nearest neighbour kernel}$$

$$\mathbf{c}_j = \text{random subset of the data points}$$

- Algorithm

$$\begin{aligned} \mathbf{c}_j(k+1) &= \mathbf{c}_j(k) + M_j(\mathbf{x}(k))[\eta(\mathbf{x}(k) - \mathbf{c}_j(k))] \\ M_j(\mathbf{x}(k)) &= \begin{cases} 1, & \text{if } v_j \|\mathbf{x} - \mathbf{c}_j\|^2 \leq v_l \|\mathbf{x} - \mathbf{c}_l\|^2 \quad \forall j \neq l \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (3.9)$$

$$\begin{aligned} v_j(k+1) &= \alpha v_j(k) + (1 - \alpha)[M_j(\mathbf{x}(k)) \|\mathbf{x} - \mathbf{c}_j\|^2] \\ H(\bar{v}_1, \dots, \bar{v}_L) &= \sum_{j=1}^L -\bar{v}_j \ln(\bar{v}_j) \quad \text{with} \quad \bar{v}_j = v_j / \sum_{l=1}^L v_l \\ \eta &= 1 - H(\bar{v}_1, \dots, \bar{v}_L) / \ln(L) \end{aligned}$$

where $M_j(\mathbf{x})$ is a membership function, η the adaptive learning rate and L the number of kernels. The adaptive k-means algorithm distributes the kernels among the clusters with equal variance. Therefore, the same bandwidth for every kernel may be used. The bandwidth may be determined by the variance of the clusters. Another approach is a P -nearest neighbour heuristic algorithm [95] to determine an appropriate bandwidth σ_j^2 .

$$\sigma_j^2 = \frac{1}{P} \sum_{i=1}^P \|\mathbf{c}_j - \mathbf{c}_i\|^2 \quad (3.10)$$

where \mathbf{c}_i are the P -nearest neighbours of \mathbf{c}_j .

The mean square error (MSE) cost function which the k-means algorithm attempts to minimize can be expressed as.

$$\text{MSE}(\mathbf{c}) = \sum_{j=1}^L v_j \quad (3.11)$$

To demonstrate the performance of the k-means clustering algorithm the Logistic map in a 2-dimensional and the Lorenz system in a 3-dimensional input space are used. In Figure 3.8(a) the initial positions of the 25 kernels are shown. After training with 10000 samples the centres of the kernels have uniformly covered the attractor in Figure 3.8(b).

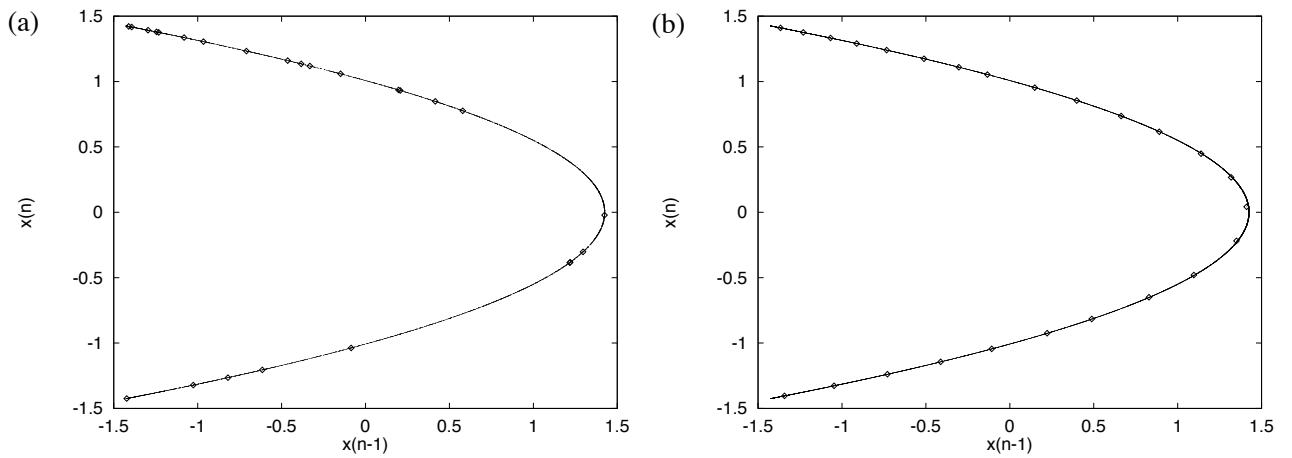


Figure 3.8: Logistic map (10000 training samples, $\alpha = 0.99$, 25 kernels) : (a) Initialisation (b) Final kernel locations

In Figure 3.9(a) the initial positions are chosen randomly from the input data. The final positions after training are shown in Figure 3.9(b). It is more difficult to tell in the 3-d plot if the kernels are positioned with an equal distance to each other. However, it can be seen that the kernels cover all the input data in a homogeneous way.

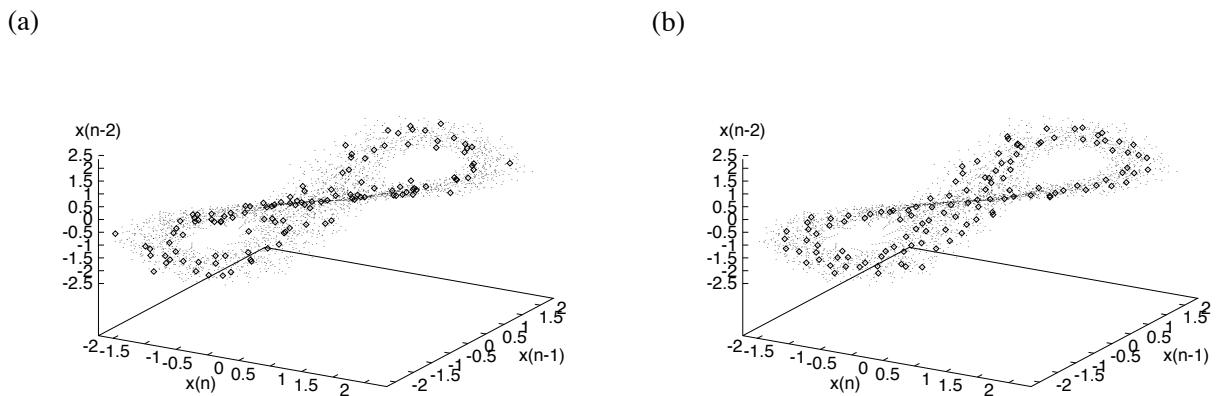


Figure 3.9: Lorenz (40000 training samples, $\alpha = 0.99$, 125 kernels) : (a) Initialisation (b) Final kernel locations

Both MSE are plotted for the 2-dimensional and 3-dimensional input spaces. The MSE in the two dimensional input space (Logistic map) is a magnitude smaller than in the 3-dimensional input space of the Lorenz data. This means that the centres in the 3-dimensional case have a greater centre variation. The adaptive k-means algorithm tries to converge to equidistant spacing between the centres. For that reason each kernel has the same bandwidth σ_i^2 . The input data in the three dimensional space is therefore not covered uniformly by the kernels and, hence, will decrease the performance of the RBF network.

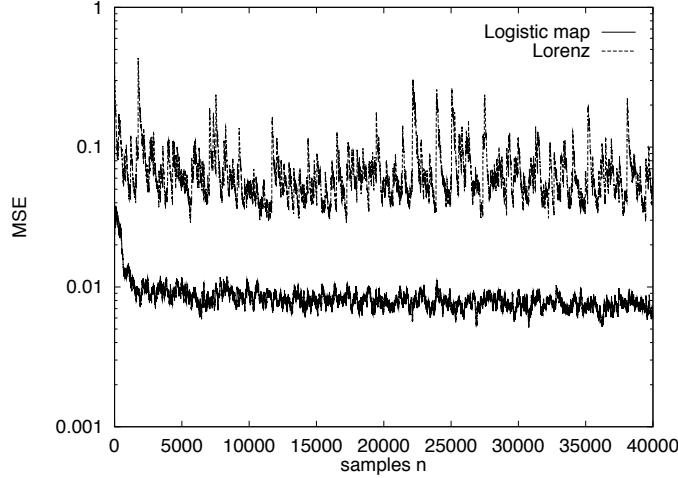


Figure 3.10: Mean square error of the variance

The maximum likelihood (ML) solution determines the location of the means of a mixture density of component Gaussian densities. The k-means algorithm is an approximate version of the ML solution. To find the exact solution the EM algorithm [112] which performs far better than the k-means algorithm [113] may be used. The EM algorithm is an iterative algorithm for a pdf estimation [114] based on a mixture of Gaussian kernels [115]. It can be used to calculate the location and the individual bandwidth of the kernels in a RBF network. The EM algorithm may be summarised as follows.

- Initialization

$$\begin{aligned}
 P(j) &= \frac{1}{L} \\
 \sigma_j^2 &= \text{distance to nearest neighbour kernel} \\
 \mathbf{c}_j &= \text{random subset of the data points}
 \end{aligned}$$

- Algorithm

$$p(\mathbf{x}_k | j) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{c}_j\|^2}{2\sigma_j^2}\right)$$

$$\begin{aligned}
 p(\mathbf{x}_k) &= \sum_{j=1}^L p(\mathbf{x}_k | j) P(j) \\
 P(j|\mathbf{x}_k) &= \frac{p(\mathbf{x}_k | j) P(j)}{p(\mathbf{x}_k)} \\
 \mathbf{c}_j &= \frac{\sum_{k=1}^N P(j|\mathbf{x}_k) \mathbf{x}_k}{\sum_{k=1}^N P(j|\mathbf{x}_k)} \\
 \sigma_j^2 &= \frac{1}{d} \frac{\sum_{k=1}^N P(j|\mathbf{x}_k) \| \mathbf{x}_k - \mathbf{c}_j \|^2}{\sum_{k=1}^N P(j|\mathbf{x}_k)} \\
 P(j) &= \frac{1}{N} \sum_{k=1}^N P(j|\mathbf{x}_k)
 \end{aligned}$$

where d is the input dimension, L the number of kernels and N the number of input vectors \mathbf{x} .

The EM algorithm is used to estimate the pdf of the Logistic map. Gaussian noise is added to the time series, generated by the Logistic map, because the single trajectory of the Logistic map causes the individual bandwidths of the kernels to converge to zero. The 2-dimensional input space is shown in Figure 3.11. The initial positions of the kernel were selected randomly from the input data. The EM algorithm finds an optimum solution after 200 iterations. It is noticeable that the bandwidths of the kernels are all too small. This behaviour has been also reported in [113]. Therefore, the bandwidths are multiplied by a constant factor β to achieve better interpolation performance in the RBF network.

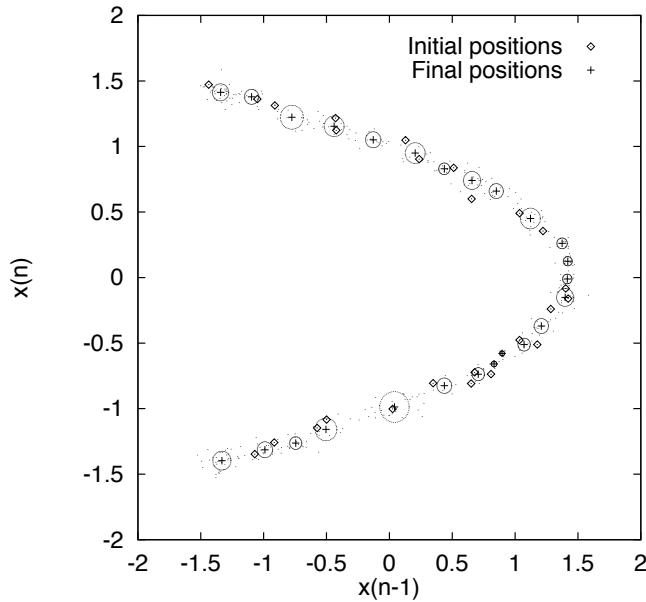


Figure 3.11: Logistic map with AWGN~ $N(0, 0.025)$, 400 data points, 200 iterations, 25 kernels

To make sure that the EM algorithm converges, the following relative error is computed.

$$\text{relative error} = \frac{\text{new value} - \text{old value}}{\text{old value}} \quad (3.12)$$

The relative errors of the bandwidth σ^2 , the prior probability $P(j)$ and the position of the kernels are plotted in Figure 3.12. It can be seen that the EM algorithm has already converged after around 80 iterations.

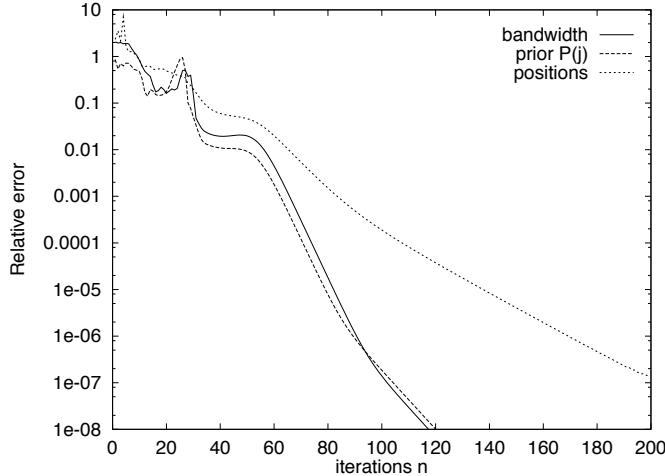


Figure 3.12: Logistic map with AWGN $\sigma^2 = 0.025$, 400 data points, 200 iterations, 25 kernels

The final positions of the kernels on the Lorenz data in a 3-dimensional input space is depicted in Figure 3.13. The initial position are selected randomly from the input data. Even though the number of data points is 4000 there are a few input samples which are separated from the main cluster. The problem with these "single" data points is that the EM algorithm positions a kernel onto that point and drives its bandwidth to zero. This causes ill-conditioning in the iterative algorithm and produces no useful results.

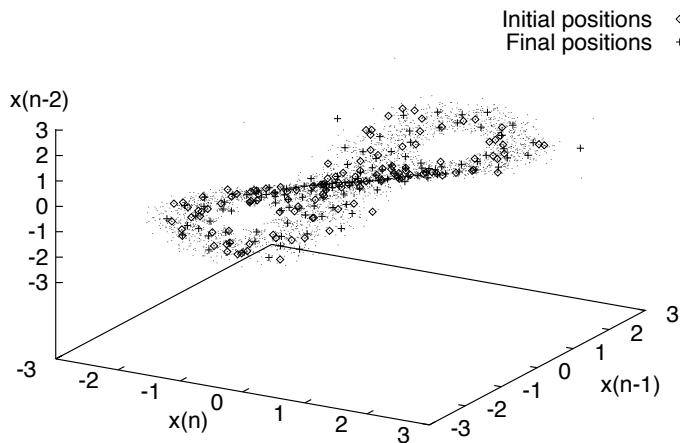


Figure 3.13: Lorenz, 4000 data points, 600 iterations, 125 kernels

The relative errors of the bandwidth σ^2 , the prior probability $P(j)$ and the position of the kernels are shown in Figure 3.14. After around 450 iterations and using 4000 data points the EM algorithm converges.

A side effect of using different bandwidths for the kernels in a NRBF is that the kernels feature multi-modal behaviour and their maximum value is not at the centre location of the kernel. These side effects may cause a

decrease in performance in a NRBF network. A further investigation of these side effects may be found in [116].

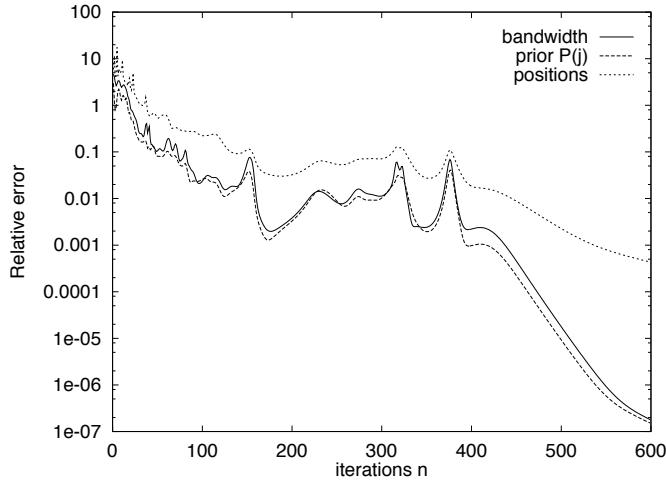


Figure 3.14: Lorenz, 4000 data points, 600 iterations, 125 kernels

The k-means clustering and the EM algorithms perform very well in 1 or 2 dimensions. The performance in higher dimensions depends on how much training data and training time is available.

For higher dimensions than 2 or 3 an evenly spread grid may be more appropriate. The dynamic range of the input data is determined and a kernel is positioned onto each of the minimum and maximum values. The other remaining kernels are distributed equally between the minimum and maximum values of the input data. The bandwidth are all the same for the kernels and are calculated by (3.10). After spanning a n-dimensional grid over the input data, the kernels which are not activated by any input vector are discarded. In Figure 3.15(a) the initial kernels on a 2-dimensional grid and in (b) the final kernel locations are shown.

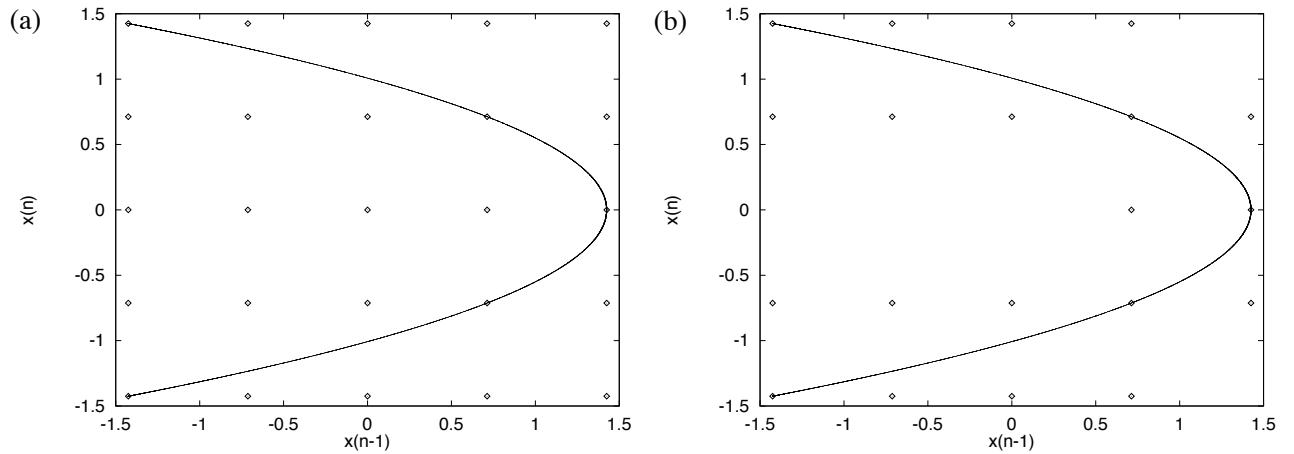
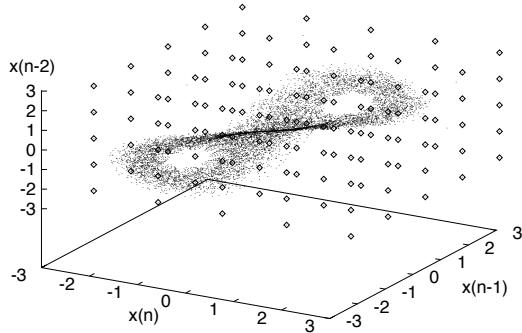


Figure 3.15: Logistic map : (a) Initialisation (b) Final kernel locations

A 3-dimensional example is shown in Figure 3.16(a) and (b). The initial 3-dimensional grid is shown in Figure 3.16(a) and the final kernel configuration in 3.16(b).

(a)



(b)

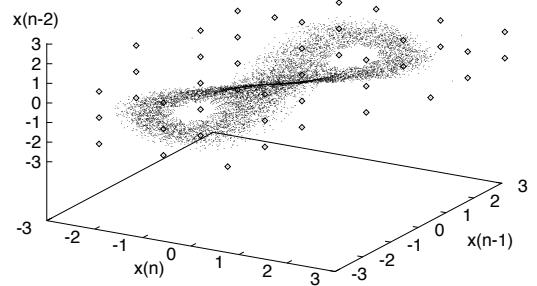


Figure 3.16: Lorenz : (a) Initialisation (b) Final kernel locations

This *ad hoc* technique usually provides a data matrix which is not ill-conditioned. Therefore, it is possible to use standard least squares techniques to determine the linear weights. The technique is also fast and reliable and is mainly used in the simulations in the following chapters.

3.3.3 Training Algorithms for the Linear Weights

Once the positions and the bandwidths for the kernels have been determined the linear weights are trained by supervised learning methods. The linear weights \mathbf{w} in the RBF network are usually trained by a least squares technique. The nonlinear expansion of the input data causes an increase in the eigenvalue spread of the auto-correlation matrix. For this reason the LMS algorithm will be very slow in convergence. The RLS algorithm is an appropriate candidate for an adaptive solution.

If the parameters of the kernels are not estimated appropriately the data matrix is often ill-conditioned, commonly caused by a too small bandwidth or a kernel location in a sparse input space. This leads to numerical problems when trying to invert the data matrix for computing the weights \mathbf{w} in the standard least squares solution. To avoid numerical instabilities singular value decomposition (SVD) of the datamatrix may be used. SVD is a block least squares technique and is computationally expensive. If the data matrix is well-conditioned an adaptive block least squares technique can be used. The Householder transformation provides an elegant solution for this kind of adaptive block least squares. The iterative algorithm may be found in Appendix A.

3.4 Chapter Summary

In this chapter two common nonlinear models were presented. The VS filter and two versions of the RBF network. The VS filter is a nonlinear model which is easy to implement and to train. Although, a low-order VS

filter might experience difficulties in modelling complex target structures, it does provide a useful benchmark for the RBF network.

The RBF network is trained in two stages. First an unsupervised learning method is used to estimate the parameters (bandwidth and centre vector) of the kernels. Secondly the linear weights are trained by a least squares technique. It was shown that the estimation of the parameters is not straightforward and, especially, in higher input dimensions the unsupervised training algorithms have great difficulties in sampling the input space correctly. Another point which arises in this context is, if it is desirable at all to position the kernels, so that they represent the input data like a probability density function. The overall aim of the RBF network is to minimise a cost function. The resulting error, however, might not be dependent on a correct statistical representation of the input space via the kernels. It is possible to tie the estimation parameters of the kernels into the minimisation process of a cost function. However, this has the disadvantage of being a nonlinear optimisation technique which exhibits local minima in the cost function.

The Householder transform is a efficient adaptive block least squares algorithm in conjunction with training the linear weights in either the VS filter or the RBF network.

Chapter 4

Interference Cancellation: Investigation of Broomhead's Filter Method

4.1 Introduction

The following two chapters examine a typical signal processing problem, which can be described as follows. A narrowband signal of interest is corrupted by wideband noise. This is a scenario which is encountered in numerous signal processing applications. Speech on a telephone line or a phase shift key (PSK) modulation scheme in a communication application are typical examples which suffer in the presence of broadband noise. An example of this scenario is shown in Figure 4.1 which shows a sketch of power spectral density (PSD) of a narrowband signal of interest, corrupted by broadband noise or interference.

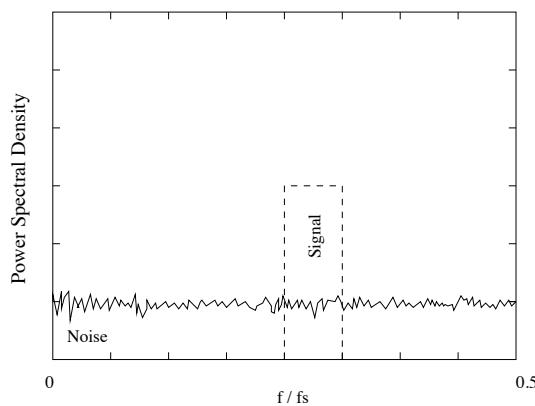


Figure 4.1: Example for a spectrum 'noise + signal of interest'

With no *a priori* knowledge about the interference or noise it is usually assumed to be a stochastic process. Conventional linear filtering techniques are then used which exploit the incoherence of the interference or spectral properties in the frequency domain. One disadvantage with these conventional techniques is that after

filtering, the band of interest is still corrupted by the noise. Linear methods cannot relate energy from one band to another. In other words, the energy of the noise outside the band of interest cannot be exploited to estimate the noise inside the band of interest. Therefore the achievable signal to noise (SNR) improvement, gained by the linear techniques, will be only sub-optimal. If the noise is purely stochastic e.g. independent identically distributed (i.i.d.), then there will be no other filtering techniques to combat the noise in the band of interest.

However, today it is known that many of these noise processes arise from nonlinear dynamical systems. Chaotic time series are a typical example of aperiodic time series which appear to be a stochastic process when analysed with second order statistics (see Chapter 2).

Deterministic nonlinear behaviour can arise from all kinds of different physical systems. Electronic circuits (Chua's circuit) [16], mechanical systems (engine noise) [15], sea clutter (fluids) [18] or ambient ocean "noise" [14] are known to exhibit chaotic behaviour. Nonlinear processes are widespread in nature. For this reason it is important to ask whether the noise process should be modelled as a stochastic or a nonlinear deterministic process. If the noise is modelled as nonlinear deterministic then nonlinear methods which depend on the coherence are more suitable to cancel nonlinear interference. In other words, nonlinear models may be used to estimate the noise in the band of interest. Therefore, an improvement in performance in the SNR may be possible when nonlinear filtering techniques are applied.

In linear signal processing the most straightforward technique to cancel noise from the corrupted signal of interest, is to subtract the modelled noise from the corrupted signal of interest. However, if the noise is nonlinear the technique of subtracting the noise from the corrupted signal of interest does not work. Consider the following example shown in Figure 4.2. Firstly, define

$x(n)$ – noise (chaotic process)

$a(n)$ – signal of interest

$y(n)$ – observed signal

The terms in the chaotic process are related by the iterative map

$$x(n+1) = f(x(n)) \quad (4.1)$$

and since f must be nonlinear to generate chaos, in general

$$f(x(n) + a(n)) \neq f(x(n)) + f(a(n)) \quad (4.2)$$

then the residual prediction error will be

$$\begin{aligned} e(n) &= x(n) + a(n) - f_{\text{filter}}(x(n-1) + a(n-1)) \\ e(n) &= f(x(n-1)) + a(n) - f_{\text{filter}}(x(n-1) + a(n-1)) \end{aligned} \quad (4.3)$$

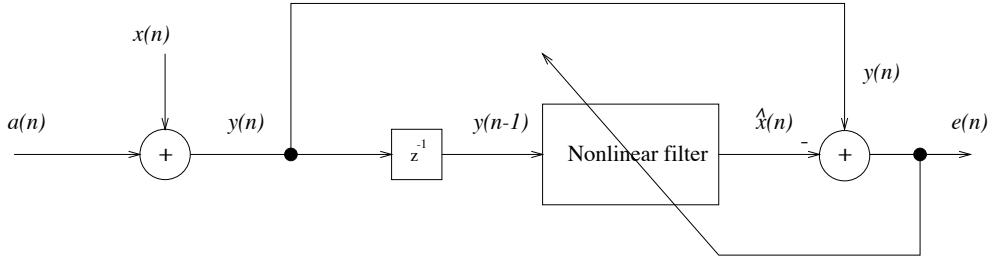


Figure 4.2: Nonlinear signal prediction

The predictor will only cancel the chaotic noise effectively, when the signal of interest $a(n)$ is very small compared to the noise $x(n)$. A modification to circumvent this restriction would be to feedback the delayed signal $\hat{a}(n - 1)$ to the nonlinear filter. This on the other hand causes the feedback to be unstable due to the chaotic noise. In [117] it is shown that the error of the estimate will grow, on average, as the largest Lyapunov exponent associated with the dynamical system, giving rise to chaotic behaviour.

Although chaotic behaviour is encountered in a variety of nonlinear systems, few researchers have addressed the problem of cancelling nonlinear deterministic noise or interference from a signal of interest. Physicists usually encounter a different dilemma. When observing a physical system, *i.e.* measuring a time series, the chaotic signal itself is the signal of interest, which is corrupted by some broadband stochastic noise. Different techniques to 'clean-up' the chaotic signal are reviewed in [9, 118, 119].

A method for the cancellation of deterministic noise from discrete signals has been developed in [120]. The method uses a predictive model with feedback to cancel the chaotic noise. It circumvents the instability problems by assuming that the signal of interest is discrete. A similar approach [121], which also uses feedback control, assumes that the signal of interest is small compared to the noise and slowly time varying. These assumptions severely limit its wider application. The method recently proposed by Broomhead *et al.* in [1], shown in Figure 4.3, replaces the restrictions on SNR with a requirement that the signal of interest should be extremely narrowband with respect to the noise. This filter method is investigated in this chapter.

In Section 4.2 the theory of the filter method for cancelling nonlinear deterministic noise, derived by Broomhead *et al.* will be investigated. Section 4.3 contains three different experiments using the filter method. In Section 4.4 practical issues are considered and simulated. A comparison with linear techniques using the same three experiments is presented in Section 4.5. Section 4.6 concludes this chapter.

4.2 Filter Method (Theory)

The filter method derived by Broomhead *et al.* [1] is a method for nonlinear signal separation. A block diagram depicting how the signal separation may be achieved is shown in Figure 4.3.

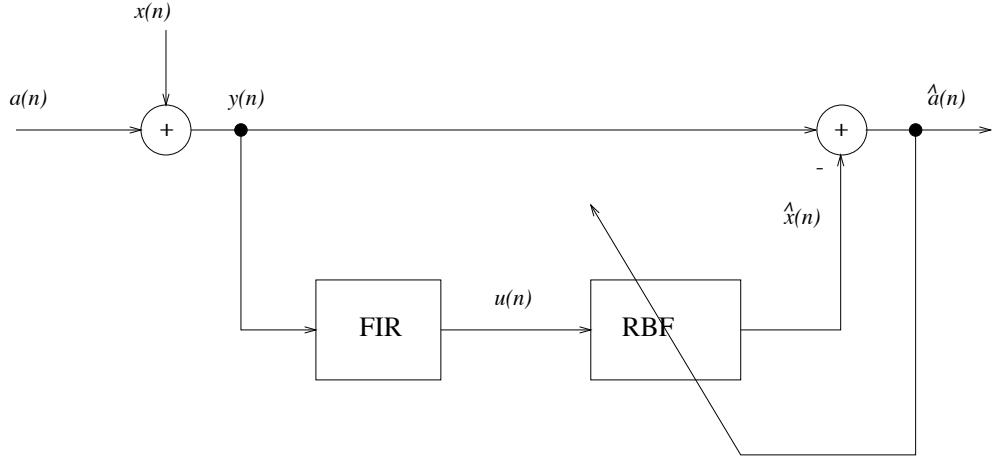


Figure 4.3: Filter method for separating signals

This approach contains no feedback, which means it will be always stable. Another advantage is that the magnitude of the signal of interest does not have to be small, compared with the magnitude of the chaotic noise. This technique exploits fundamental properties of delay embedding and assumes a knowledge of the spectral bandwidth of the signal of interest.

The task of the linear FIR filter is to cancel the signal of interest $a(n)$ from the corrupted time series $y(n)$ without spoiling the dynamics of the chaotic process $x(n)$. Since FIR filters do not have a feedback path, these filters preserve the structure of a dynamic process [4]. Infinite impulse response (IIR) filters, for instance, have a feedback path and it has been shown [122, 123] that these filters with their own dynamics increase the dimension of the chaotic signal. Recently, it has been suggested [124] that if certain conditions are fulfilled, an IIR filter might also preserve the dynamical structure of a chaotic noise process. Although FIR filters do give an embedding of the original system, the usage of a high order FIR filter may change the dynamics of a chaotic process, as pointed out in [4].

The FIR filter is a bandstop filter, which should ideally be orthogonal to the signal of interest, *i.e.*

$$\sum_{k=0}^{N-1} h(k)a(n-k) = 0 \quad \forall n \quad (4.4)$$

Where $h(k)$ are the coefficients of the FIR filter and $a(n - k)$ is the sampled signal of interest. A delay map $\mathbf{F}_N \Psi_{L+N-1}$ can be constructed from the output $u(n)$ of the FIR filter. Ψ_{L+N-1} is the embedding of M using $(L + N - 1)$ delays using real-valued measurements of the unfiltered time series $\{y(n)\}$

$$\Psi_{L+N-1} = [y(n), y(n-1), \dots, y(n-L-N+1)]^T \quad (4.5)$$

M is the manifold of the dynamical system. \mathbf{F}_N is an $L \times (L + N - 1)$ banded matrix

$$\mathbf{F}_N = \begin{bmatrix} h(0) & h(1) & h(2) & \cdots & h(N-1) & 0 & 0 & \cdots & 0 \\ 0 & h(0) & h(1) & \cdots & h(N-2) & h(N-1) & 0 & \cdots & 0 \\ 0 & 0 & h(0) & \cdots & h(N-3) & h(N-2) & h(N-1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & h(0) & h(1) & h(2) & \cdots & h(N-1) \end{bmatrix}. \quad (4.6)$$

\mathbf{F}_N describes the effect that filtering has on the embedded attractor. The filtered delay map is shown in Figure 4.4

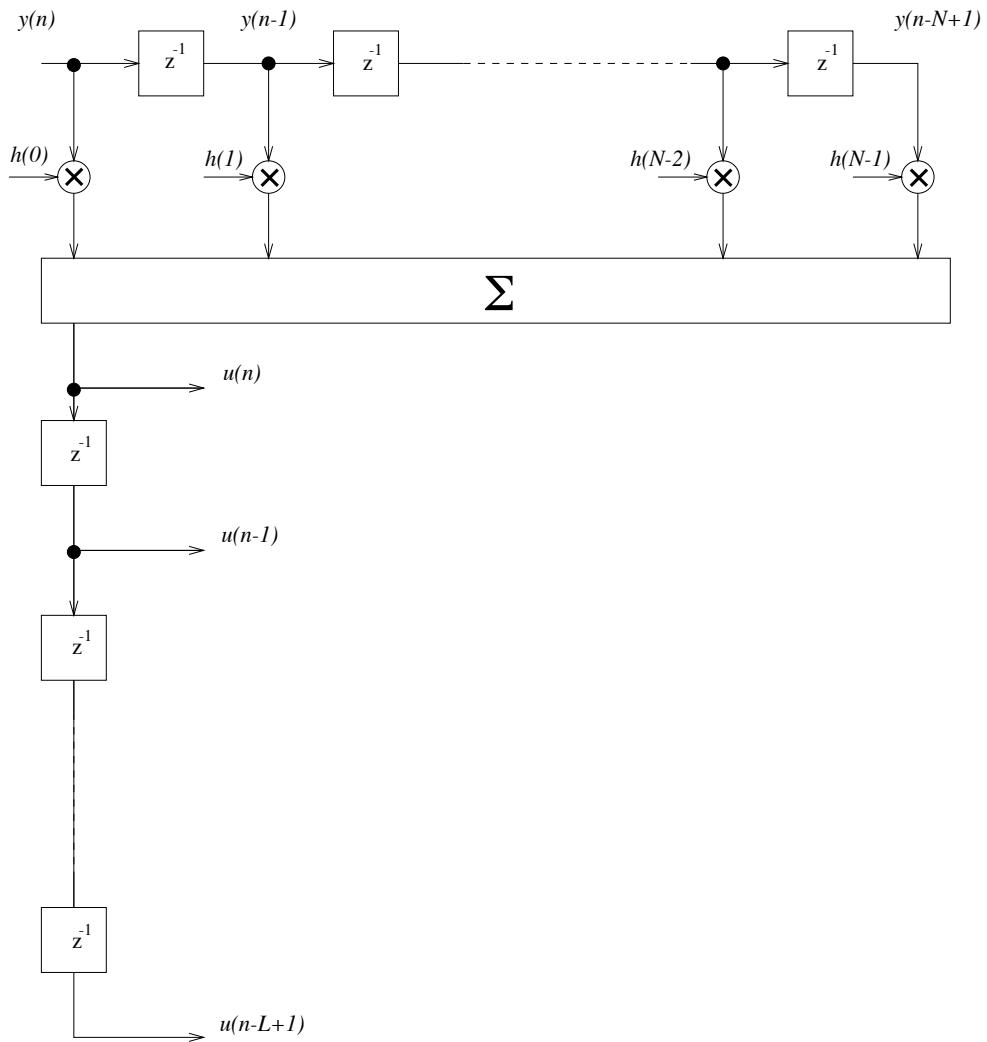


Figure 4.4: Delay embedding with a filtered time series

The aim of the nonlinear filter is to reconstruct the dynamic structure of the chaotic noise. The linear map \mathbf{F}_N has cancelled the signal of interest and distorted the image of the chaotic attractor. An inverse \mathbf{F}_N^{-1} has to be estimated by the nonlinear filter $\widehat{\mathbf{F}}_N^{-1}$, in order to undo the distortion. Assuming that \mathbf{F}_N cancels the signal of

interest $a(n)$ completely

$$\mathbf{F}_N \mathbf{y} = \mathbf{F}_N \mathbf{x} \quad (4.7)$$

an estimate of the chaotic noise $x(n)$ is

$$\hat{\mathbf{x}} = \hat{\mathbf{F}}_N^{-1}(\mathbf{F}_N \mathbf{y}) \quad (4.8)$$

Once the estimated map $\hat{\mathbf{F}}_N^{-1}$ has been found the message $a(n)$ can be estimated as follows

$$\hat{a}(n) = y(n) - (\hat{\mathbf{F}}_N^{-1}(\mathbf{F}_N \mathbf{y})) \quad (4.9)$$

To fit the data in the form of points $(\mathbf{F}_N(\mathbf{y}), \mathbf{y})$, which lie close to the ideal graph of \mathbf{F}_N^{-1} , a nonlinear model, for instance a normalised RBF network, may be implemented.

A valid question to ask at this point would be, if it is really necessary that the inverse of the linear FIR filter has to be nonlinear. First of all, an inverse filter to a FIR filter is an IIR filter. Since the FIR filter used in the filter method is a notch filter with a zero on the unit circle, its inverse is an ill-defined IIR filter. Another point is that a linear inverse filter is defined for every input. This is not necessary when estimating $\hat{\mathbf{F}}_N^{-1}$. The attractor of the dynamical system does not occupy the whole state space. This is also in strong contrast to a stochastic process. Therefore it is only necessary to invert the points on the distorted attractor, which is only possible with a nonlinear model.

4.3 Theory: Simulations and Results

In order to investigate and to verify the theory for the filter method described in the previous section, three experiments have been carried out. These three experiments have been also carried out in [1] and, although, no quantitative results are given in [1] the results can be generally confirmed.

Experiment I :

A sine wave is corrupted by broadband chaotic noise. The chaotic noise is generated from the complex Ikeda map. It derives from Maxwell's equations for a plane wave propagating in an optical ring cavity.

$$z(n+1) = 1 + \mu z(n) e^{j(\alpha - \frac{\beta}{1+|z(n)|^2})} \quad (4.10)$$

The real part of the complex Ikeda map $x(n) = \Re\{z\}$ is used to generate the chaotic noise series. In order to generate chaotic behaviour the parameters are chosen to be $\mu = 0.7$, $\alpha = 0.4$ and $\beta = 6.0$. The signal of interest $a(n)$ is a sine wave with an amplitude of 0.2. The normalised frequency f/f_s of the sine wave is 0.28125. 7500 samples were generated and split into three sets, as described in Chapter 3. A NRBF network and a VS filter described in Chapter 3 are used to estimate the nonlinear map $\hat{\mathbf{F}}_N^{-1}$. The embedding dimension is $L = 4$. 94 centers from a 4-dimensional grid for the NRBF network are chosen (see Chapter 3). The bandwidth of each

center was $\sigma^2 = 0.25$. To create a notch filter [125] the following equation is used

$$H(z) = G(1 - 2 \cos(\omega_o)z^{-1} + z^{-2}) \quad (4.11)$$

hence

$$\begin{aligned} h(0) &= 1.0 \\ h(1) &= -2 \cos(2\pi f/f_s) = 0.3902 \\ h(2) &= 1.0 \end{aligned}$$

The amplifying factor is $G = 1.0$. The power spectral density (PSD) of the corrupted signal $y(n) = a(n) + x(n)$ is depicted in Figure 4.5(a). The input SNR is -2.7 dB. After passing the corrupted signal $y(n)$ through the FIR filter the PSD changes as in Figure 4.5(b). It can be seen that the FIR filter completely removes the sine wave.

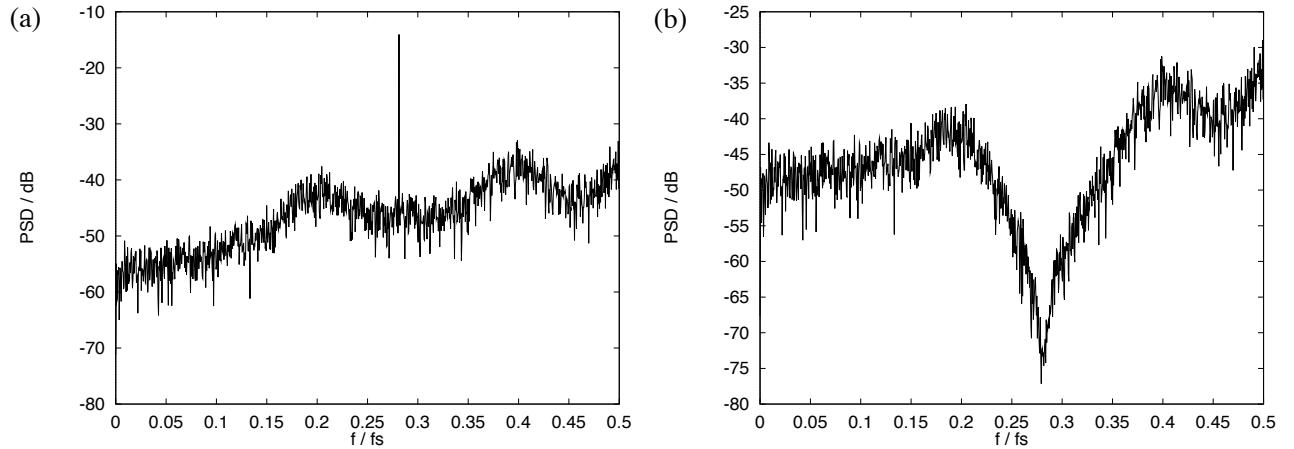


Figure 4.5: PSD : (a) Corrupted signal $y(n)$ (b) Filtered signal $u(n)$

To demonstrate the effects of the filtering process on the chaotic time series $x(n)$ the attractors before and after filtering are shown in Figure 4.6(a) and (b), respectively. Although there are some additional twists and shifts in the filtered attractor, there is still some resemblance to the original attractor.

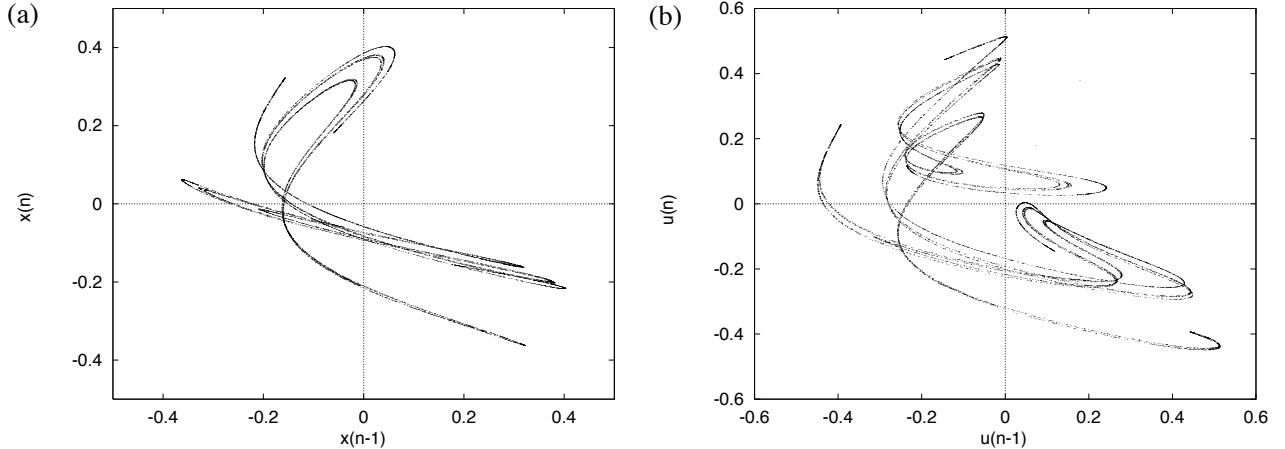


Figure 4.6: Phase plane : (a) Original attractor (b) Filtered attractor

After applying the filter method, the achieved output SNR is 7.5 dB for both nonlinear models. If it is possible to train the nonlinear model without the signal of interest present, hence with only the chaotic noise, the output SNR can be further improved by around 1 dB. This further improvement holds for both nonlinear models. The PSD of the recovered signal of interest $\hat{a}(n)$ is shown in Figure 4.7. It can be seen that the chaotic noise around the sine wave is attenuated and, just as important, the sine wave still has its original energy.

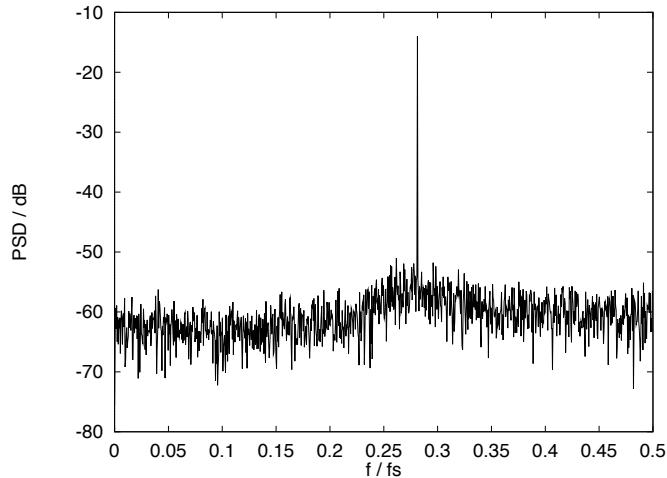


Figure 4.7: PSD of recovered signal of interest

A convergence plot of the achieved normalised mean square error (NMSE) of the NRBF network can be seen in Figure 4.8. The NMSE is computed, as follows :

$$\text{NMSE} = 10 \log_{10} \left(\frac{\sigma_{\hat{a}}^2}{\sigma_y^2} \right) \quad \text{dB} \quad (4.12)$$

where $\sigma_{\hat{a}}^2$ is the variance of $\hat{a}(n)$ and σ_y^2 is the variance of $y(n)$.

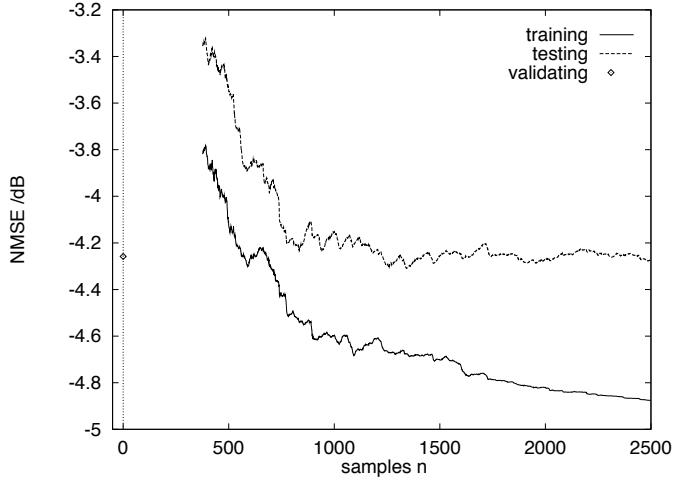


Figure 4.8: NMSE in dB for training, testing and validating

In Figure 4.8 it can be seen that the training and the testing error converge, which means that the NRBF network is not overfitted (see Chapter 3). The validation error is similar to the testing error, which also demonstrates the good generalisation performance of the NRBF network. The reconstructed attractor of the Ikeda map can be seen in Figure 4.9.

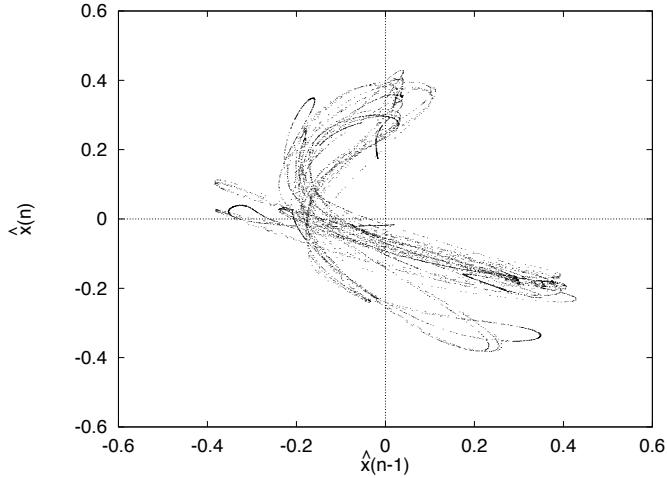


Figure 4.9: Reconstructed attractor of the Ikeda map

Although the reconstructed attractor is a bit fuzzy, it can be seen that the nonlinear model successfully counteracts the distortions caused by the FIR filter. There is resemblance between the original, in Figure 4.6(a), and the reconstructed attractor of the time series $\hat{x}(n)$ in Figure 4.9.

Experiment II :

In this experiment a phase shift key (PSK) modulated sine wave is corrupted by chaotic noise from the Lorenz

system [9]. The three nonlinear differential equations were solved by the Runge-Kutta method [126].

$$\begin{aligned}\frac{dx(t)}{dt} &= \sigma(y(t) - x(t)) \\ \frac{dy(t)}{dt} &= -x(t)z(t) + rx(t) - y(t) \\ \frac{dz(t)}{dt} &= x(t)y(t) - bz(t)\end{aligned}\quad (4.13)$$

The parameters σ , r and b are set to be 10, 28 and $8/3$, respectively. The phase of the sine wave is randomly chosen at every 30th sample to be 0 or π . The sine wave's frequency is $f/f_s = 0.17$ and its amplitude is 0.1. A FIR filter with a notch at $f/f_s = 0.17$ cancelled the carrier sine wave. The PSD of the Lorenz noise $x(n)$ and of the signal of interest $a(n)$ are shown in Figure 4.10. The input SNR is -23 dB.

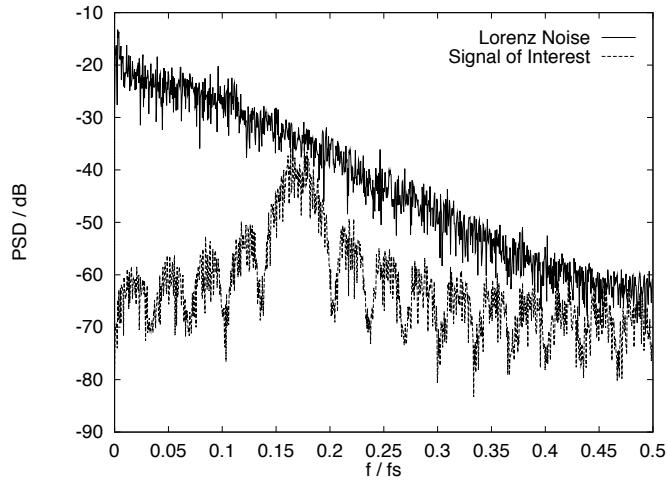


Figure 4.10: PSD of signal of interest and Lorenz noise

The corrupted signal $y(n)$ can be seen in Figure 4.11(a). It is clear that the signal of interest $a(n)$ cannot be identified. Figure 4.11(b) shows the PSD after filtering the corrupted signal.

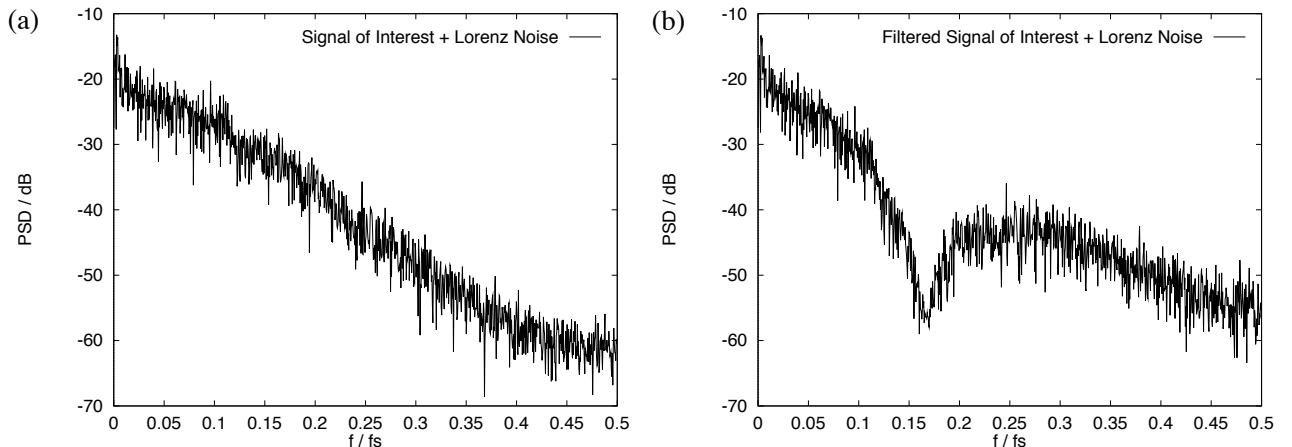


Figure 4.11: PSD : (a) Corrupted signal $y(n)$ (b) Filtered signal $u(n)$

A NRBF network with 174 kernels, each with a bandwidth of 1.8, is trained with 2500 samples. The testing and validation set also contained 2500 samples each. The achieved output SNR is -0.5 dB. The original modulated sine wave, together with its recovered version and the binary signal, which is scaled down by factor 2 for displaying purposes, can be seen in Figure 4.12(a).

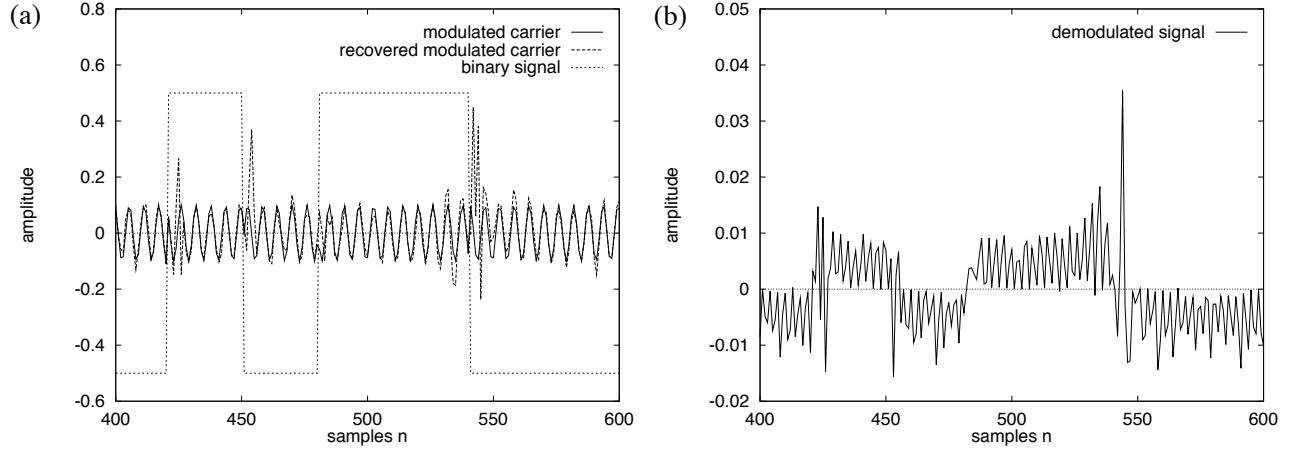


Figure 4.12: Time series : (a) Corrupted signal $y(n)$ and recovered signal $\hat{a}(n)$ (b) Demodulated signal

Every time the phase of the sine wave changes, a portion of the sine waves leaks through the FIR filter and produces relatively big errors in the recovered signal. Figure 4.12(b) shows the demodulated signal of interest. The demodulation is achieved by multiplying the recovered modulated carrier by a sine wave with the same frequency and phase. The next step would be to lowpass filter the demodulated signal to obtain the binary signal. The convergence performance of the achieved NMSE of the NRBF network is shown in Figure 4.13.

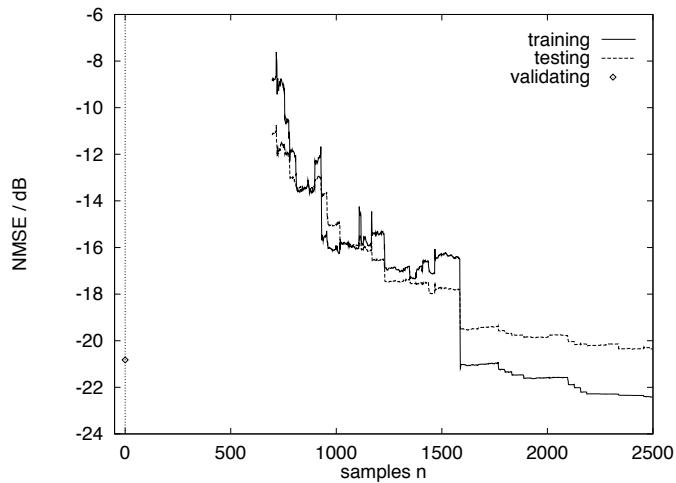


Figure 4.13: NMSE in dB for training, testing and validating

In Figure 4.13 it can be seen that the training and the testing error converge, which means that the NRBF network is not overfitted. The validation error is similar to the testing error, which also shows good generalisation of the NRBF network.

Experiment III :

Two chaotic time series are used in this experiment. The Rössler and the Lorenz chaos. Both are added up to form the corrupted signal $y(n)$, where the Rössler chaos is chosen to be the signal of interest. The differential equations which generate the Rössler time series are as follows

$$\begin{aligned}\frac{dx(t)}{dt} &= -(y(t) + z(t)) \\ \frac{dy(t)}{dt} &= x(t) + ay(t) \\ \frac{dz(t)}{dt} &= b + z(t)(x(t) - c)\end{aligned}\quad (4.14)$$

The parameters are $a = 0.398$, $b = 2.0$ and $c = 4.0$. The input SNR is 0 dB and the achieved output SNR, with the training sequence, is 3.1 dB. This experiment was far harder to deal with because a big portion of the signal of interest was leaking through the FIR filter. Training the NRBF network poses problems, *e.g.* spikes in the output $\hat{x}(n)$ and generally a poor performance with the test and validating signals. The PSD of the Rössler chaos is depicted in Figure 4.14(a). Most of its energy is located on the frequency bin $f/fs = 0.08$.

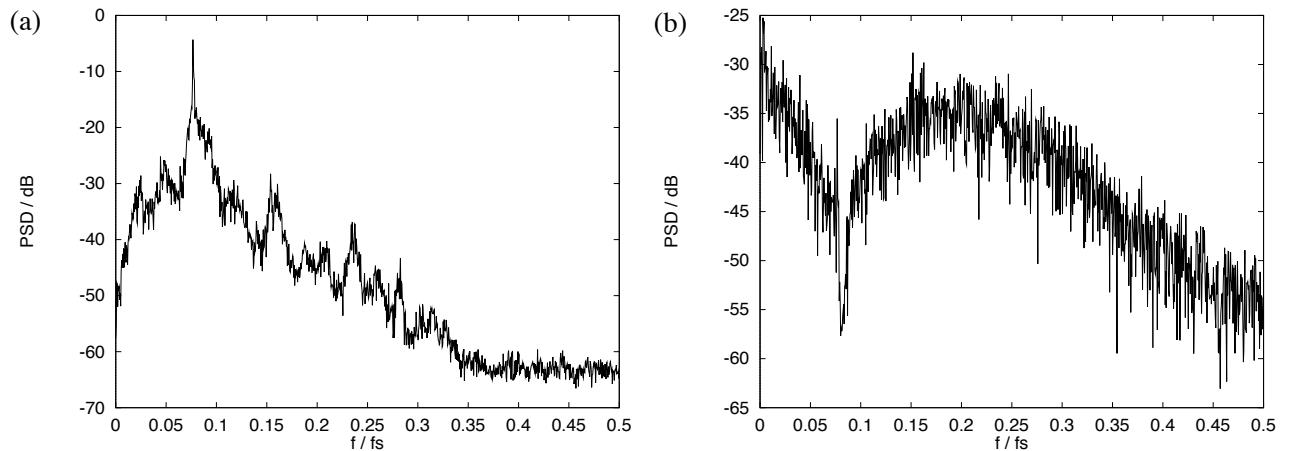


Figure 4.14: PSD : (a) Rössler chaos $a(n)$ (b) Filtered Lorenz + Rössler $u(n)$

Therefore, an FIR with a notch at $f/fs = 0.08$ is chosen to cancel most of the Rössler chaos. The resulting PSD is shown in Figure 4.14(b).

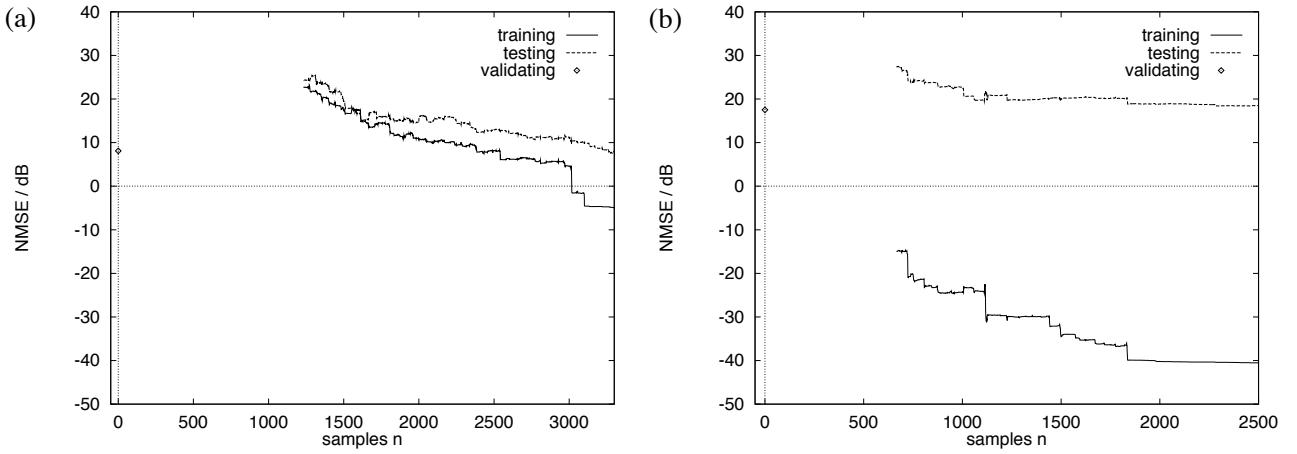


Figure 4.15: NMSE in dB for training, testing and validating : (a) Training with the signal of interest present
 (b) Training without the signal of interest present

As mentioned earlier this experiment proves more challenging. This can be noticed in Figure 4.15(a) and (b). In Figure 4.15(a) all three sets contained the signal of interest. Only the NMSE of the training sequence is acceptable. The NMSE of both the testing and the validation set show that the NRBF network fails to model the nonlinear inverse of the linear FIR filter. When the signal of interest is not present, as shown in Figure 4.15(b), the training error shows good performance. The testing and the validation sets, with the signal of interest present, produce unacceptable results.

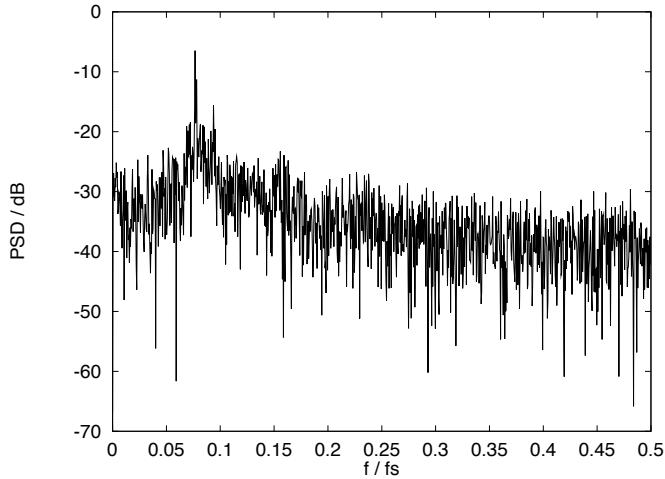


Figure 4.16: PSD of the recovered Rössler chaos

In Figure 4.16 the PSD of the recovered Rössler chaotic time series $\hat{a}(n)$ is shown. In comparison with the PSD of the original Rössler chaos shown in Figure 4.14(a) it can be seen that the energy on the frequency bin $f/fs = 0.08$ has been successfully recovered, but also a considerable amount of noise is still present above the frequency $f/fs = 0.08$.

4.4 Practice: Simulations and Results

In experiment II and III it could be seen that the performance of the filter method decreases significantly, when the FIR filter is not able to cancel the signal of interest completely. One clear drawback with this filter method is, that the FIR filter has to be orthogonal to the signal of interest in order to perform well. Usually the signal of interest is more complicated than a sine wave, which was the case in experiment I.

To demonstrate the effect of using a conventional bandstop filter experiment I is carried out again. This time the sine wave $f/f_s = 0.28125$ is cancelled by a bandstop filter with a band start frequency $f/f_s = 0.26$ and a bandstop frequency $f/f_s = 0.3$. Two FIR filters are designed, one with an attenuation of -6 dB with 25 coefficients and one with an attention of -50 dB with 193 coefficients. The attractors of the filtered time series $u(n)$ are shown in Figure 4.17 (a) using 25 coefficients and in (b) using 193 coefficients.

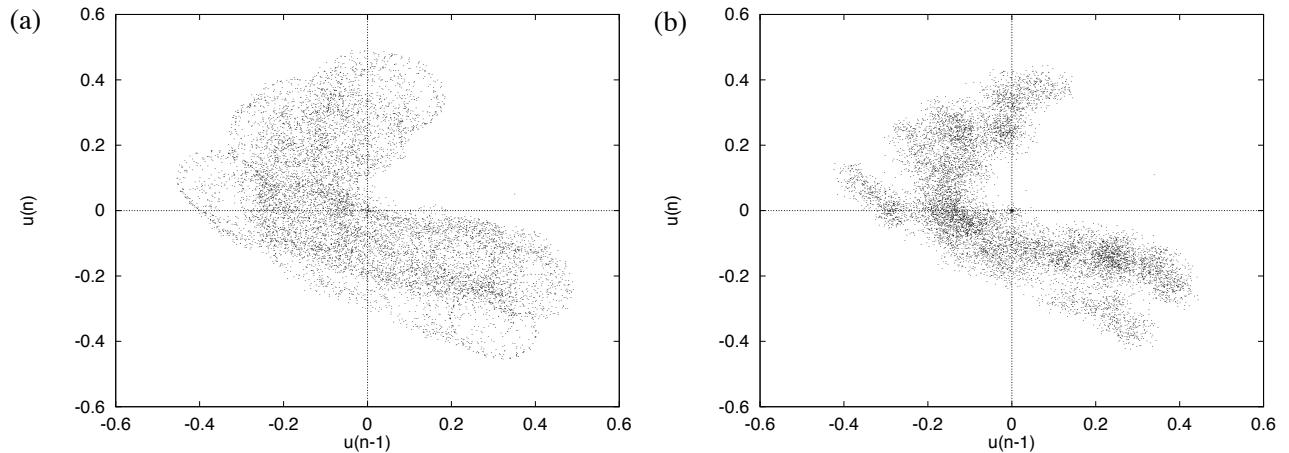


Figure 4.17: Attractors with different FIR filters : (a) 25 coefficients (b) 193 coefficients

The attractors are heavily blurred. It is very difficult for a nonlinear model to undo the distortions caused by the linear FIR filters. Although the sine wave does not seem to appear in Figure 4.17(b) the original attractor of the Ikeda map is almost unrecognisable.

Furthermore the sine wave is still present in the input of the nonlinear model and, therefore, will also be modelled by the nonlinear model, as shown in Figure 4.18. This decreases the output SNR.

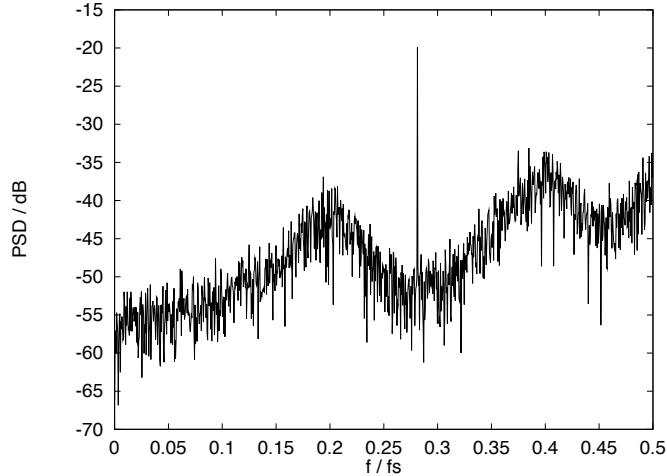


Figure 4.18: PSD of the filtered time series $u(n)$ of the nonlinear model

A NRBF network with 447 centres ($\sigma^2 = 0.36$) is trained with 2500 samples. The FIR filter with 193 coefficients is used to cancel the sine wave. The PSD of the output of the NRBF network is shown in Figure 4.19(a) and the PSD of the recovered signal of interest is shown in Figure 4.19(b). It is clear that the NRBF network fails completely in this scenario.

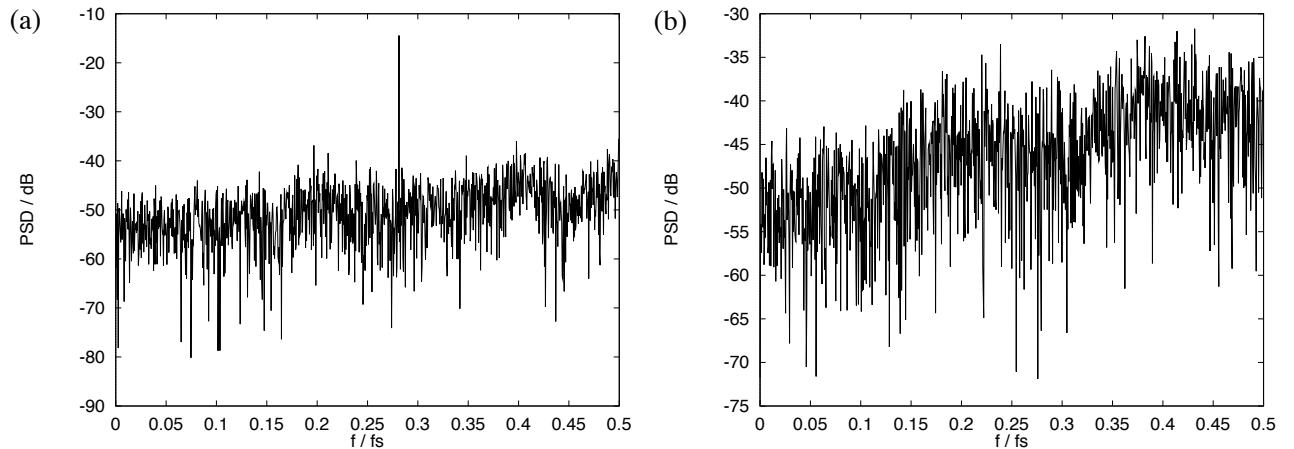


Figure 4.19: PSD : (a) Output of the NRBF network (b) Recovered signal of interest

Another practical issue is, that to a certain degree stochastic noise will be present in the signal of interest. This will also blur the attractor and cause fitting problems for any nonlinear model.

4.5 Comparison with Linear Techniques

The output SNRs in the previous section showed an improvement compared to the input SNRs in the experiments I - III. Those positive results were achieved with nonlinear models. To gain further insight the same three

experiments were carried out, but this time with linear filtering techniques. Nonlinear models are computationally far more expensive than linear ones. For this reason it is important to re-investigate the experiments in order to justify the use of nonlinear filters. As the signals of interest are narrowband, one obvious linear approach is to use a bandpass filter. The bandpass filters are designed to capture the narrowband of interest. To verify the claim that the inverse of the linear FIR filter has to be nonlinear, a linear adaptive FIR filter [127–129] is used in the filter method instead of the nonlinear model. Both linear approaches deal with the same experiments as in the previous section.

4.5.1 Simulations and Results

Experiment I :

A linear bandpass IIR filter is used to filter the sine wave $f/fs = 0.28125$. The bandpass is a 6th order Butterworth filter with normalised startband frequency $f/fs = 0.28$ and stopband frequency $f/fs = 0.2825$. From an input SNR of -2.7 dB the achieved output SNR is 22.9 dB. The PSD of the recovered sine wave is depicted in Figure 4.20.

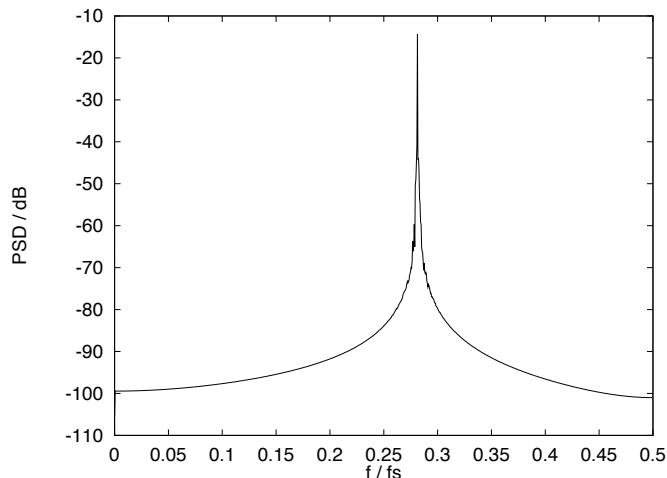


Figure 4.20: PSD of the recovered signal of interest

To recover a single sine wave from broadband noise is a straightforward task for the IIR filter. The 6th order Butterworth IIR filter [37] produces a very sharp bandpass filter, which leaves little chaotic noise in the band of interest.

The other scheme uses a linear adaptive filter instead of a nonlinear model in the filter method. An adaptive filter with 10 coefficients has been trained with 1100 samples. The PSD of the recovered sine wave is shown in Figure 4.21(a).

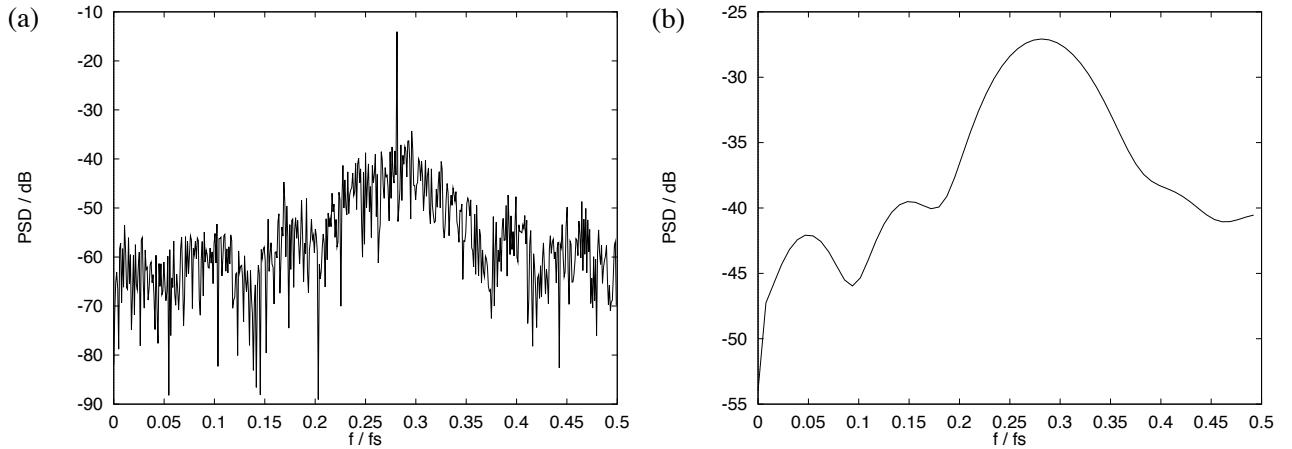


Figure 4.21: PSD : (a) Recovered signal of interest (b) Transfer function of the linear adaptive filter

The transfer function, after convergence in the least square sense, of the linear adaptive filter is shown in Figure 4.21(b). The achieved output SNR is 6.6 dB. The achieved result is almost as good as with the nonlinear model.

Experiment II :

A linear bandpass IIR filter is used to capture the carrier sine wave $f/f_s = 0.17$. The 2nd order Butterworth filter has a start band frequency $f/f_s = 0.168$ and stop band frequency $f/f_s = 0.172$. The achieved SNR is -2.3 dB. The original modulated sine wave, together with its recovered version and the binary signal, which is scaled down by factor 2 for displaying purposes, can be seen in Figure 4.22.

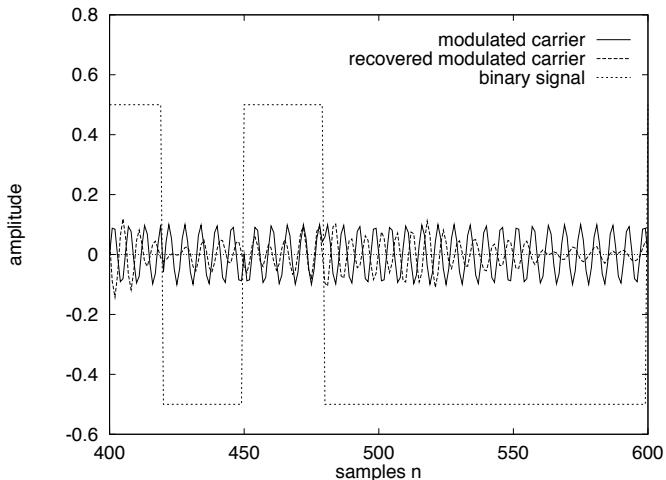


Figure 4.22: Time series

Although only a 2nd order IIR filter is used the IIR filter starts oscillating for a relatively short time period when the phase changes, causing the output SNR to decrease.

Again, a linear adaptive filter with 10 coefficients was trained with 1100 samples. The recovered modulated

sine wave and the original modulated sine wave are shown in Figure 4.23(a). Neither linear filter performs as well as the filter method with the nonlinear model.

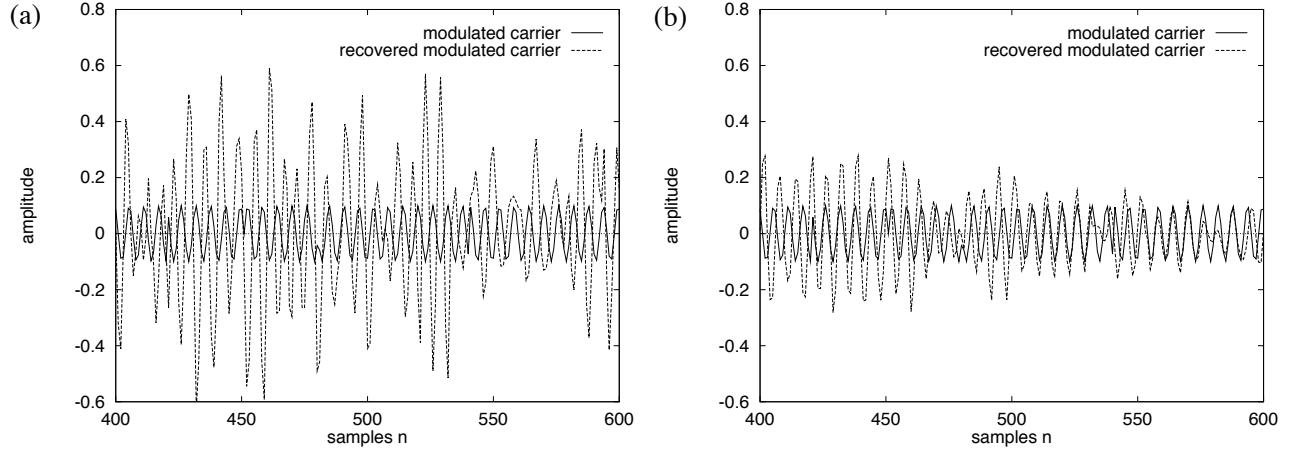


Figure 4.23: Time series (a) Linear adaptive filter with 10 coefficients (b) Linear adaptive filter with 50 coefficients

The linear adaptive filter achieves an output SNR -11.9 dB. The filter clearly does not perform as well as the nonlinear model in the previous section. A linear adaptive filter with 50 coefficients was trained on 5000 samples. It achieved a better performance with an output SNR of -6.1 dB. The recovered modulated sine wave and the original modulated sine wave are shown in Figure 4.23(b).

Experiment III :

In this experiment a 2nd order Butterworth IIR filter is used to capture the main harmonic of the Rössler chaos. The IIR filter is a bandpass filter with a start frequency at $f/f_s = 0.06$ and a stop frequency $f/f_s = 0.1$. The achieved output SNR is 4.4 dB.

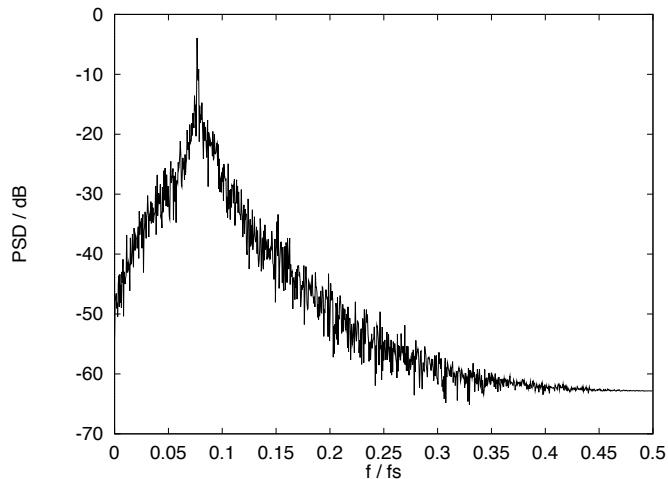


Figure 4.24: PSD of the recovered Rössler chaos using a bandpass filter

To improve the SNR filterbanks could be used to capture the other significant harmonics of the Rössler chaos. The other linear scheme used a linear adaptive FIR filter with 10 coefficients. It was trained with 1100 samples. The PSD of the recovered Rössler chaos is shown in Figure 4.25(a).

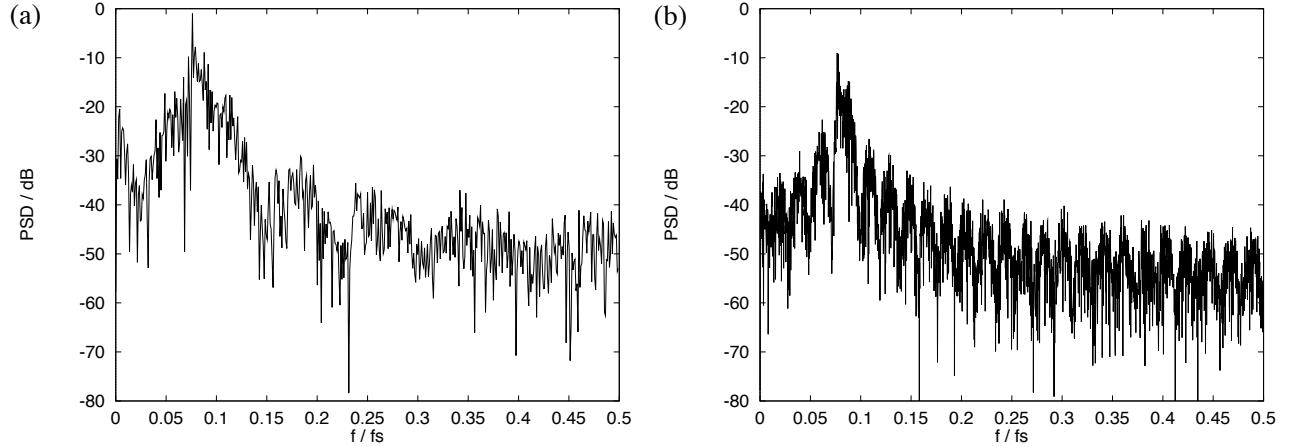


Figure 4.25: PSD of the recovered Rössler chaos (a) Linear adaptive filter with 10 coefficients (b) Linear adaptive filter with 50 coefficients

The achieved output SNR is -4.2 dB. This linear filter used in the filter method does not improve the input SNR of 0 dB. A linear adaptive filter with 50 coefficients was trained with 5000 samples. The PSD of the recovered Rössler chaos is shown in Figure 4.25(b). The achieved output SNR is 1.7 dB. The filter method with the nonlinear model achieved 3.1 dB.

Table 4.1 summarises all the results of experiments I-III.

Experiment	SNR			
	Original	Nonlinear	IIR filter	FIR filter
I	-2.7	7.5	22.9	6.6
II	-23	-0.5	-2.3	-6.1
III	0	3.1	4.4	1.7

Table 4.1: SNR in *dB* for the different linear and nonlinear filters

4.6 Chapter Summary

In this chapter, Broomhead et al.'s filter method was investigated. On first inspection the filter method seems to be a promising approach to cancel nonlinear deterministic noise from a narrowband signal of interest. The major drawback with this method is that the FIR filter has to be orthogonal to the signal of interest. This is only possible if the signal is a pure sine wave. If the requirement of orthogonality is not met, the performance of the filter method decreases significantly. This is because the signal of interest leaks through the FIR filter and is modelled by the nonlinear model as well as the chaotic noise. The signal of interest in the corrupted time

series $y(n)$ is reduced by the output $\hat{x}(n)$ of the nonlinear model.

On the other hand, if a high order FIR filter, with a strong attenuation in the band of interest is used, the structure of the dynamical system becomes heavily blurred. Estimating \mathbf{F}_N^{-1} becomes very difficult and simulations have shown that the NRBF network fails.

The comparison with linear filtering techniques in experiments I - III show that it is possible to get satisfactory results using conventional linear bandpass filters. This fact is especially important, because the computational effort of training a nonlinear model is far greater than using an IIR filter. The use of a linear adaptive filter, instead of a nonlinear model, in the filter method showed generally a performance decrease. This confirms the theory of the filter method that the optimum inverse map \mathbf{F}_N^{-1} has to be nonlinear.

Nevertheless, the principle of modelling nonlinear deterministic noise in the band of interest with nonlinear models is very appealing. The following chapter contains modifications to the filter method which enables the filter method to be used in a more practical environment.

Chapter 5

Interference Cancellation: Modification and Re-embedding

5.1 Introduction

In this chapter three different schemes for cancelling nonlinear deterministic noise are derived and investigated. All three methods are based on the filter method of the previous chapter whose main conclusion was that the filter method is not suitable to be applied in a practical environment. For this reason the filter method has been re-examined and modified.

The first method presented in Section 5.2 is the new filter method. It exploits standard signal processing techniques [130], i.e. filterbanks, multirate signal processing and linear adaptive filtering, as well as the nonlinear deterministic nature of the chaotic noise. In Section 5.3 another method is introduced which uses a quantiser. The quantisation noise from the quantiser masks the signal of interest at the input to the nonlinear model. The third method, discussed in Section 5.4 investigates the scenario where three sets of FIR filters and NRBF networks are cascaded. The task of the cascaded sets is to recover three sine waves from nonlinear deterministic noise. The main purpose of this scheme is to break up high order FIR filters into sequential sets of very short ones. Section 5.5 concludes this chapter.

5.2 New Filter Method

In the previous chapter it was shown that the performance of the filter method of Broomhead *et al.* degrades heavily when the FIR filter is not orthogonal to the signal of interest. There are two reasons for this reduction in performance, which depend on the order (length) of the FIR filter. Firstly, if the FIR filter (not orthogonal to the signal of interest) is of low order¹ the signal of interest is not attenuated very much. Therefore, the

¹Estimate from experiments

nonlinear model is able to estimate the signal of interest, as well as the chaotic noise. Hence, the output of the nonlinear model also contains the signal of interest, which degrades the output SNR.

If a high order FIR filter is used, it is possible to achieve a strong attenuation in the band of interest. However, the dynamics become heavily distorted. It becomes very difficult for the nonlinear model to estimate the non-linear inverse of the linear FIR filter.

To circumvent this dilemma, a new filter method is proposed. To illustrate the procedure of the the new filter method the example I, with slight changes, from the previous chapter has been used. A block diagram of the new filter method is shown in Figure 5.1.

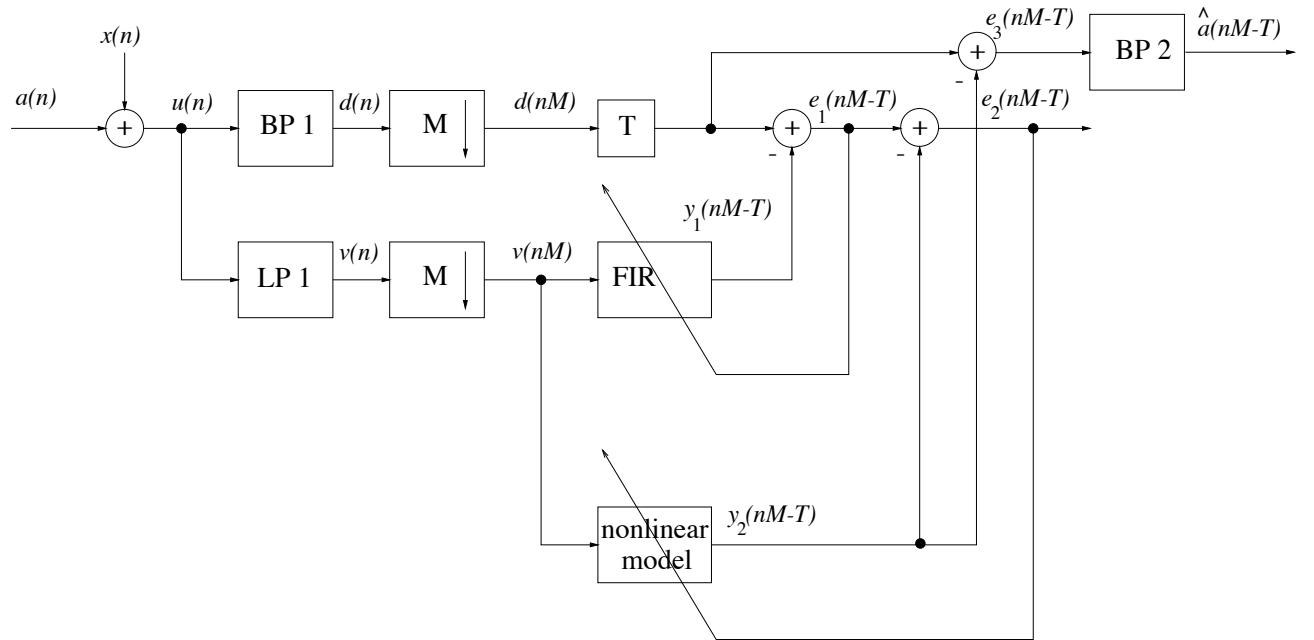


Figure 5.1: Nonlinear interference cancellation

The signal of interest $a(n)$ is a sine wave with a frequency of $f/f_s = 0.125$. The chaotic noise is generated by the Lorenz system as before. The PSD of the corrupted sine wave $u(n)$ can be seen in Figure 5.2.

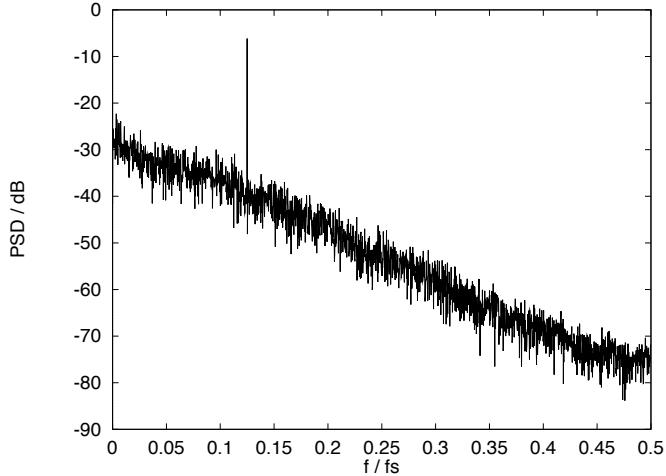


Figure 5.2: Sine wave corrupted by chaotic noise derived from the Lorenz system

5.2.1 FIR Filterbanks

The task of the linear FIR lowpass filter LP1, in Figure 5.1, is to provide the nonlinear model with a reference signal $v(n)$. The aim is to attenuate the sine wave as much as possible, without significantly distorting the dynamic structure of the chaotic noise. In this example a FIR filter with 25 coefficients is chosen. The PSD of the lowpass filtered signal $v(n)$ can be seen in Figure 5.3(a). The sine wave has been attenuated by around 15 dB. The chaotic noise is passed through the LP1 filter up to $f/fs = 0.1$ without attenuation.

It is not necessary for the nonlinear model to estimate the whole bandwidth of the chaotic noise since the bandwidth of the noise in the region of the signal of interest, is sufficient. Therefore, a bandpass filter is designed to isolate the band of interest. The bandpass filter in this example has 25 coefficients. The bandpass filter BP1 provides the desired signal $d(n)$ for the linear adaptive FIR filter. The PSD of the bandpass filtered signal $d(n)$ is depicted in Figure 5.3(b).

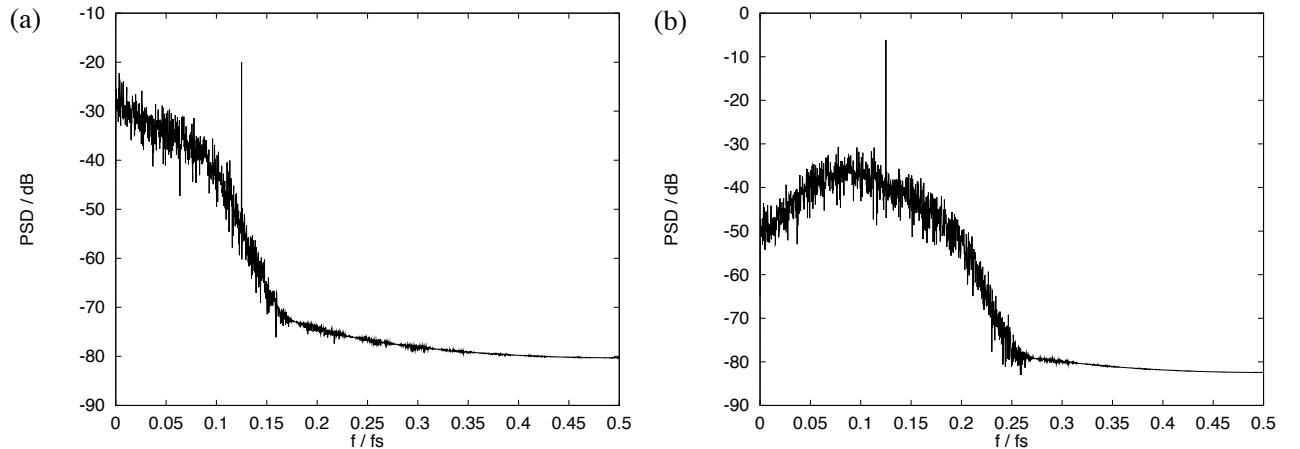


Figure 5.3: PSD : (a) Lowpass filtered signal $v(n)$ (b) Bandpass filtered signal $d(n)$

5.2.2 Multirate Signal Processing

As the two signals $d(n)$ and $v(n)$ are now bandlimited it is possible to decimate the signals [131] by the integer factor M effectively re-embedding the filtered chaotic signal. Decimating the signals also makes the scheme computationally more efficient because the nonlinear model has a smaller size, and is implemented at a lower rate [132].

There are two ways to determine the downsampling factor M . A linear approach arises from the fact that aliasing in the frequency domain has to be avoided [133, 134]. The new sampling rate is defined as

$$F' = \frac{1}{T'} = \frac{1}{MT} = \frac{F}{M} \quad (5.1)$$

Assuming a digital lowpass filter that approximates the following ideal characteristics,

$$H_L(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \frac{2\pi F' T}{2} = \frac{\pi}{M} \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

the sampling rate reduction is then achieved by saving only every M th sample of the filtered output. The bandpass filter BP1 is not ideal and has a start frequency at $f/f_s = 0.05$ and a stop frequency at $f/f_s = 0.2$. Therefore, to avoid aliasing a suitable downsampling factor is $M = 2$.

In order to exploit the determinism and structure of the chaotic noise, it is important to fully unfold the chaotic attractor [66]. To estimate an appropriate sampling rate a nonlinear approach may be used. Average mutual information (see Chapter 2) measures the independence between adjacent samples with a varying time lag T . In Figure 5.4 the average mutual information $I(T)$ is plotted for the chaotic noise, lowpass filtered chaotic noise and the bandpass filtered chaotic noise. It has been suggested [66] that the first minimum is a good indicator for choosing an appropriate sampling rate. The first minimum of the Lorenz data occurs at $T = 0.02$. The filtered versions have a slight information increase and their minimas lie at around $T = 0.025 \dots 0.03$. As the Lorenz data is sampled at 100 Hz it is appropriate to downsample the data by factor $M = 2$.

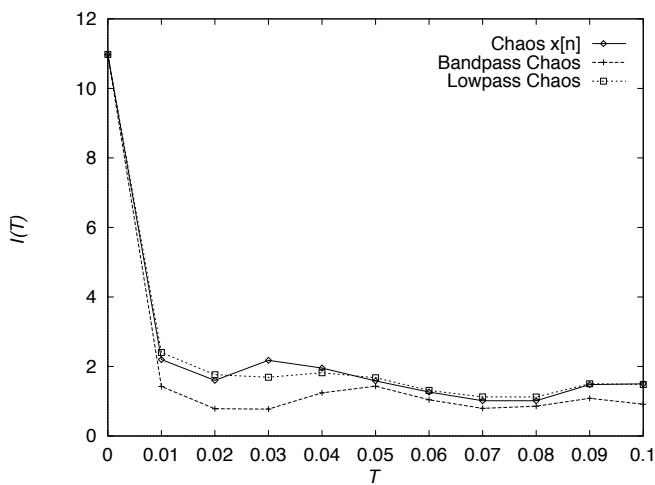


Figure 5.4: Average mutual information $I(T)$

Both linear and nonlinear approaches indicate that a downsampling factor $M = 2$ is most appropriate. The PSD of the downsampled signals $v(nM)$ and $d(nM)$ can be seen in Figure 5.5(a) and (b), respectively.

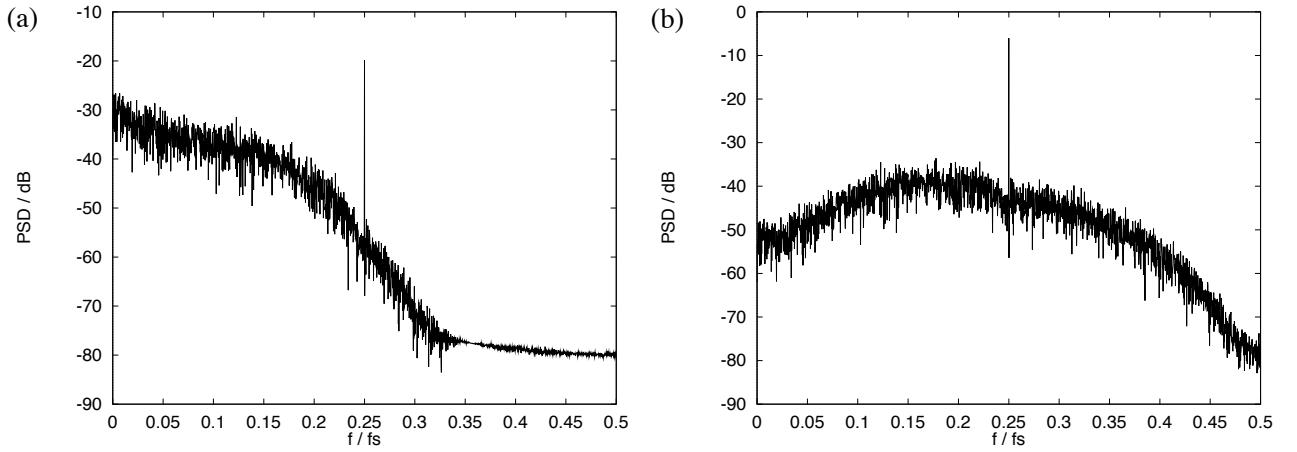


Figure 5.5: PSD : (a) Downsampled signal $v(nM)$ (b) Downsampled signal $d(nM)$

To restore the original frequency an interpolator, as seen in Figure 5.6, is used. The sampling rate is increased by an integer factor L . The new sampling period, T' is

$$\frac{T'}{T} = \frac{1}{L} \quad (5.3)$$

Therefore, the new sampling rate F' is

$$F' = LF \quad (5.4)$$



Figure 5.6: Interpolation and bandpass filtering

The interpolator inserts $L - 1$ zeros between each sample. In this case, $M = 2$, one zero is inserted between every sample. The same bandpass BP1 can be used to cancel the images of the baseband centered at harmonics of the original sampling frequency.

5.2.3 Adaptive Linear Filter

In general the signal of interest, $a(n)$, will be attenuated by LP1 but not removed. Therefore, the nonlinear model will also try to model the signal of interest as well as the chaotic noise. To avoid this, it is necessary to remove the linear map between $v(nM)$ and $d(nM)$. Assuming that the attenuated signal of interest can be

estimated by a linear model, the following approach has been devised. A linear adaptive FIR filter is trained to model the linear map. After convergence in the least square sense the error signal $e_1(nM)$ will be orthogonal to the input vector to the filter. This means that there is no linear map between $v(nM)$ and $e_1(nM)$. The PSD of the error signal $e_1(nM)$ of an adaptive linear filter with 4 coefficients and a delay of $T = 4$ is depicted in Figure 5.7. The optimum delay T was estimated experimentally and the number of coefficients correspond to the embedding dimension of the nonlinear filter. The adaptive filter is trained with a recursive least squares (RLS) [127] algorithm.

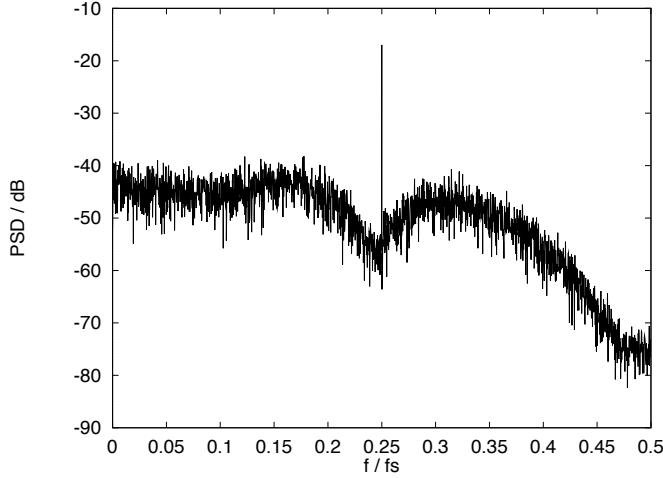


Figure 5.7: Error signal $e_1(nM)$ of the linear adaptive FIR filter

5.2.4 Nonlinear Model

A nonlinear model is used to estimate the nonlinear mapping between the signals $v(nM)$ and $e_1(nM)$. The output of the nonlinear model $y_2(n)$ is the estimated nonlinear noise. Therefore, an estimate of the signal of interest can be found by $e_3(nM) = d(nM) - y_2(nM)$.

The bandpass filter BP2 is necessary because the nonlinear model introduces unwanted high frequency components outside the bandwidth of interest.

5.2.5 Simulations and Results

To compare the normalised RBF network with another nonlinear model, a VS filter was chosen. The VS filter is a 3rd order (linear, quadratic and cubic terms) polynomial filter. A linear filtering technique, which is a combination of bandpass BP1, decimation by factor M and followed by bandpass BP2 is provided as a benchmark for comparison. The linear scheme can be seen in Figure 5.8.

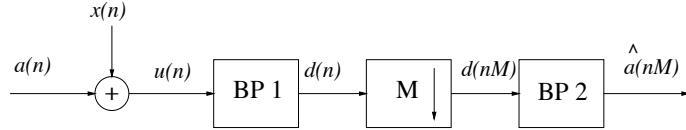


Figure 5.8: Linear filtering

The chaotic noise, as mentioned earlier, is derived from the three nonlinear differential equations (4.14) of the Lorenz system. The signal of interest is a sine wave with a frequency of $f/f_s = 0.125$. The PSD of the input signal $a[n] + x[n]$ is shown in Figure 5.2. Figures 5.9 to 5.12 show the achieved output SNR versus the input SNR across a range from -5 dB to 5 dB for different embedding dimensions.

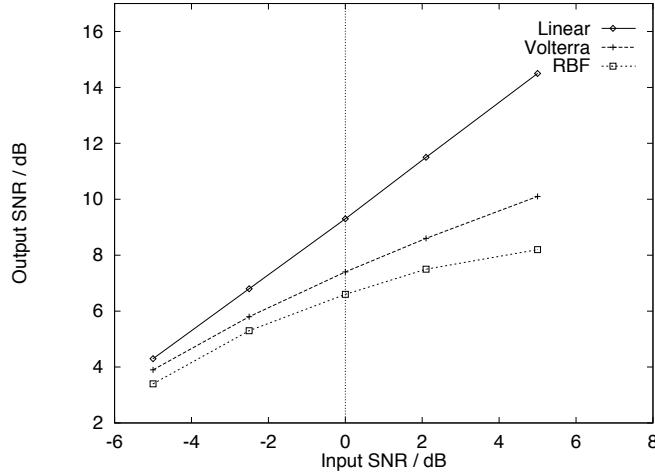


Figure 5.9: Input and output SNR in dB with an embedding dimension of 2

In Figure 5.9 the embedding dimension is 2. The NRBF network has 70..74 centres (-5dB..5dB), and was trained with 520 samples. Testing is done with a different set of 520 samples. The Volterra filter has 9 weights and was trained with 520 samples. Another set of 520 samples was used to test the Volterra filter. The results shown in Figure 5.9 are from the testing samples. The linear filtering technique achieves the best results over the whole input range.

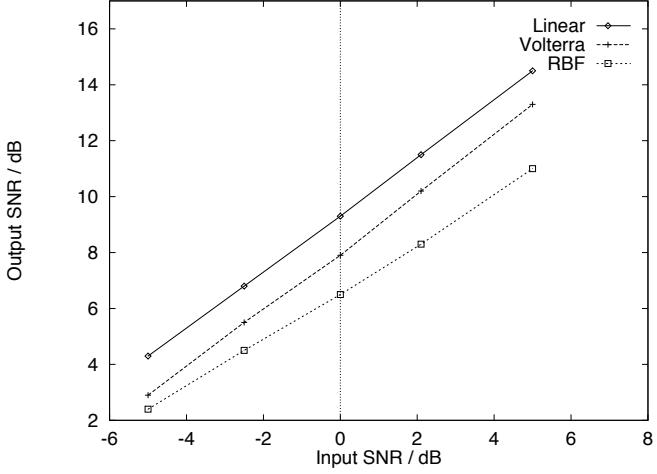


Figure 5.10: Input and output SNR in dB with an embedding dimension of 3

The results for an embedding dimension of 3 are shown in Figure 5.10. The NRBF network has 290..363 centres (-5dB..5dB). It was trained with 1200 samples and tested with a different set of 1200 samples. The Volterra filter has 19 weights and was trained with 520 samples. Another set of 520 samples was used to test the Volterra filter. The achieved output SNRs of both nonlinear models are below those achieved by the linear filtering technique. Figures 5.9 and 5.10 show that if the distorted chaotic noise is embedded in too small a dimension, the nonlinear models will not be able to exploit the determinism in the chaotic noise and, therefore, fail to estimate the original chaotic noise.

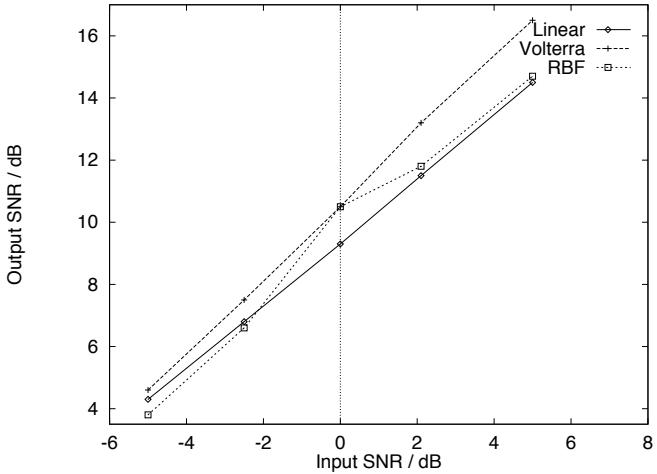


Figure 5.11: Input and output SNR in dB with an embedding dimension of 4

The situation changes, when the embedding dimension is 4. The NRBF network has 786..988 centres (-5dB..5dB). It was trained with 2500 samples and tested with a different set of 2500 samples. The Volterra filter has 34 weights and it was trained with 520 samples. Another set of 520 samples was used to test the Volterra filter. Both nonlinear models achieve a better performance compared to the linear filtering technique. This is depicted in Figure 5.11.

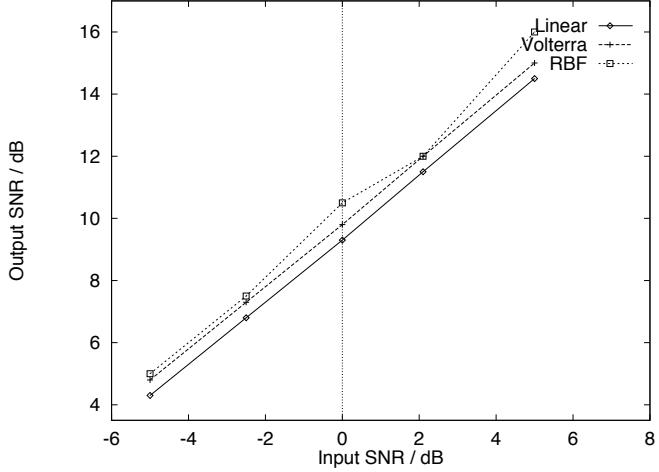


Figure 5.12: Input and output SNR in dB with an embedding dimension of 5

In Figure 5.12 the embedding dimension is 5. The NRBF network has 1508..1742 centres (-5dB..5dB). It was trained with 4200 samples and tested with a different set of 4200 samples. The Volterra filter has 55 weights and it was trained with 520 samples. Another set of 520 samples was used to test the Volterra filter. In this case the NRBF network achieves the best result. The linear filtering technique, of course, does not benefit from the right embedding dimension and performs only sub-optimally. The nonlinear models achieve up to 2 dB improvement in the output SNR compared to the linear filtering technique.

The next step is to investigate how the new filter method copes with a narrowband signal of interest, which is obtained by passing white Gaussian noise through an IIR filter. The IIR filter is a 6th order Butterworth filter designed to be a bandpass with a start frequency $f/f_s = 0.1$ and a stop frequency of $f/f_s = 0.15$. Again, the chaotic noise is derived from the Lorenz system.

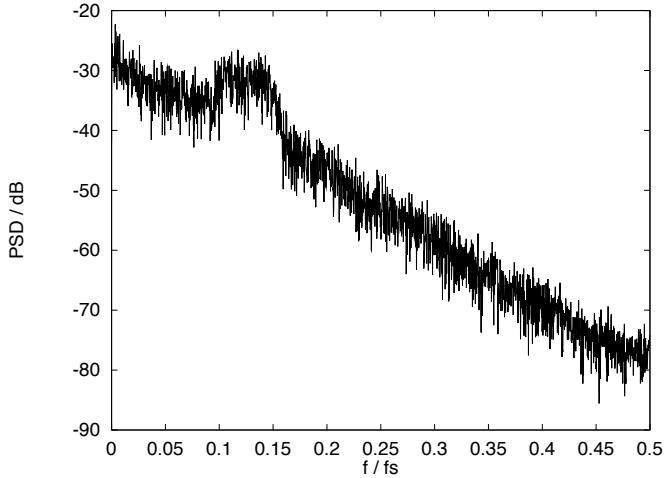


Figure 5.13: PSD : Corrupted narrowband signal of interest in chaotic noise

The PSD of the corrupted narrowband signal of interest in chaotic noise can be seen in Figure 5.13. The results are obtained using an embedding dimension of 5, but with different lengths of filterbanks (LP1 and BP1), and

are shown in Figure 5.14.

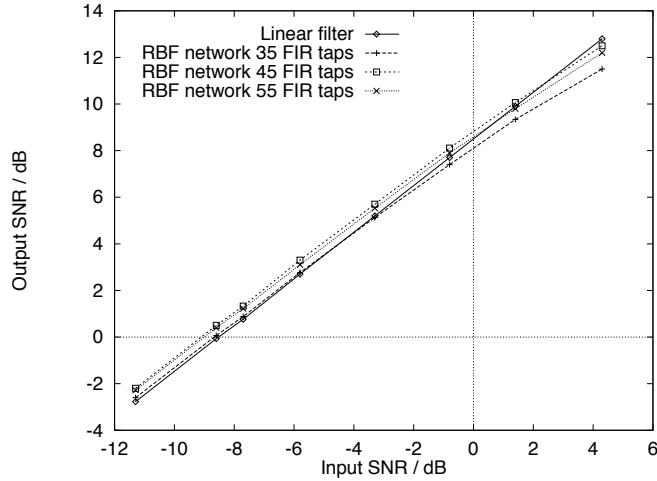


Figure 5.14: Input and output SNR in dB for different FIR filter lengths

The NRBF network does not achieve a significant improvement in the output SNR however, an interesting result is evident. The filterbanks with 45 coefficients compared with the ones with 35 and 55 coefficients marginally achieve the best result. This is evidence for the claim made earlier in Section 5.2, that there is a tradeoff between low and high order FIR filters.

5.2.6 Discussion

The restrictive assumption of the previous chapter, that the FIR filter had to be orthogonal to the signal of interest, has been removed by employing additional signal processing techniques. It could be seen that it is possible to achieve up to 2 dB improvement in performance with the new filter method, compared to the linear filtering technique. In order to achieve a better performance it is important to chose the right embedding dimension. Although, better results would have perhaps been achieved with an embedding dimension of 6, the "curse of dimensionality" prevents the design of a NRBF network that size.

However, the new filter method struggles to achieve a significant improvement in performance compared to linear techniques, when the signal of interest is more complex. The results show that the length of the FIR filters LP1 and BP1 affect the performance of the new filter method.

5.3 Quantisation

The previous chapter showed that if, in the filter method, the signal of interest leaks through the FIR filter, the performance of the filter method significantly goes down. To prevent the nonlinear model estimating the

signal of interest as well as the chaotic noise another scheme has been devised. A quantiser is introduced into the filter method, as shown in Figure 5.15. The lowpass LP1 and the bandpass BP1 have the same task as in the previous section. The quantiser is located after the lowpass LP1 and the decimator. The task of the quantiser is to mask the signal of interest with its own quantisation noise. The quantisation noise is uncorrelated with the signal of interest. Therefore, the nonlinear model is not able to estimate the signal of interest.

Section 5.3.1 gives an introduction to the quantisation scheme. In Section 5.3.2 the quantiser is designed and Section 5.3.3 presents some simulations and results. The discussion takes place in Section 5.3.4.

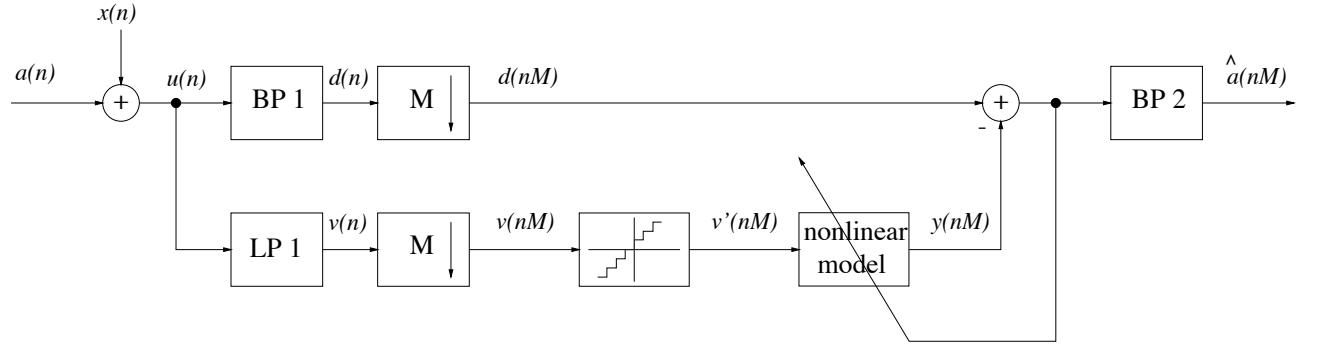


Figure 5.15: Nonlinear interference cancellation using a quantiser

5.3.1 Design

To mask the signal of interest with quantisation noise the energy of the quantisation noise has to be greater than the energy of the signal of interest. In order to estimate the energy in the signal of interest, the signal of interest on its own is filtered by the lowpass LP1. After that the variance of the signal is estimated. The variance of the quantisation noise can be estimated as follows [135].

$$\sigma_q^2 = \frac{1}{3}v_{\max}^2 2^{-2R} \quad (5.5)$$

where R is the number of bits/sample and v_{\max} is the maximum value of the input signal of the quantiser. The number of quantisation levels L is determined by

$$L = 2^R \quad (5.6)$$

Hence, to obtain the necessary quantisation levels L , given the quantisation noise σ_q^2 and the maximum value of the input v_{\max} (assuming zero mean and uniform quantisation) the following equation is used

$$L = \sqrt{\frac{v_{\max}^2}{3\sigma_q^2}} \quad (5.7)$$

5.3.2 Simulations and Results

A sine wave with an amplitude of 0.03 and frequency $f/f_s = 0.125$ is corrupted by chaotic noise. The chaotic noise is derived from the Lorenz system, with the same parameters as in the last section. A PSD plot of the corrupted sine wave can be seen in Figure 5.16. The input SNR is -27 dB. To compare the performance two simulations were carried out. The first simulation relies only on the filterbanks and decimation, *i.e.* without a quantiser. The second simulation utilises a quantiser to mask the attenuated signal of interest.

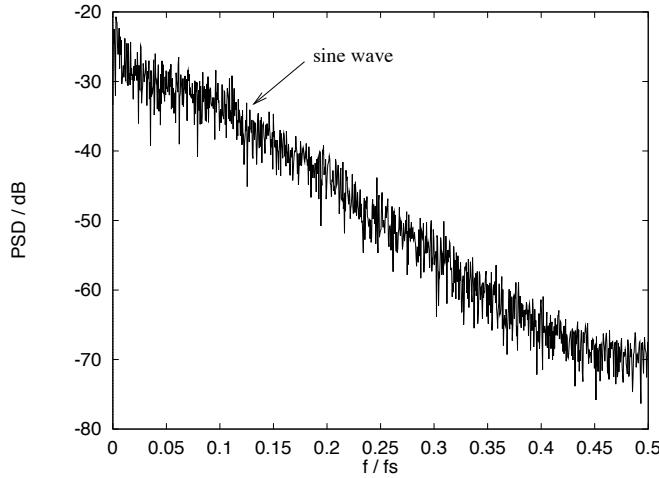


Figure 5.16: PSD : Sine wave in chaotic noise

Bandpass BP1 has 25 coefficients, a start frequency at $f/f_s = 0.05$ and a stop frequency at $f/f_s = 0.2$. The lowpass filter LP1 also has 25 coefficients with a cut-off frequency is at $f/f_s = 0.1$. To mask the signal of interest, after lowpass filtering and downsampling by $M = 2$, 8 quantisation levels are chosen. This results in quantisation noise of $\sigma_q^2 = 1.8 \times 10^{-3}$. The variance of the signal of interest after lowpass filtering is $\sigma_s^2 = 1.8 \times 10^{-5}$. A portion of the signal $v(nM)$ and its quantised version $v'(nM)$ is shown in Figure 5.17.

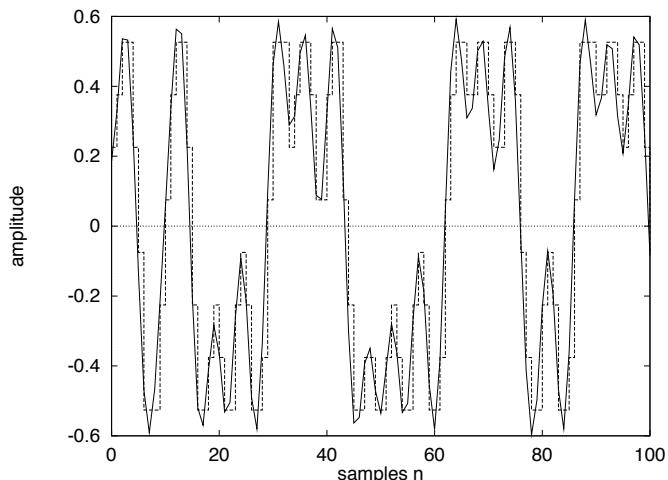


Figure 5.17: Time series : Signal $v(nM)$ and quantised signal (dotted) $v'(nM)$

The PSD of the signal $v(nM)$ and of its quantised version $v'(nM)$ is shown in Figure 5.18. It is noticeable that the sine wave has now been masked by the quantisation noise.

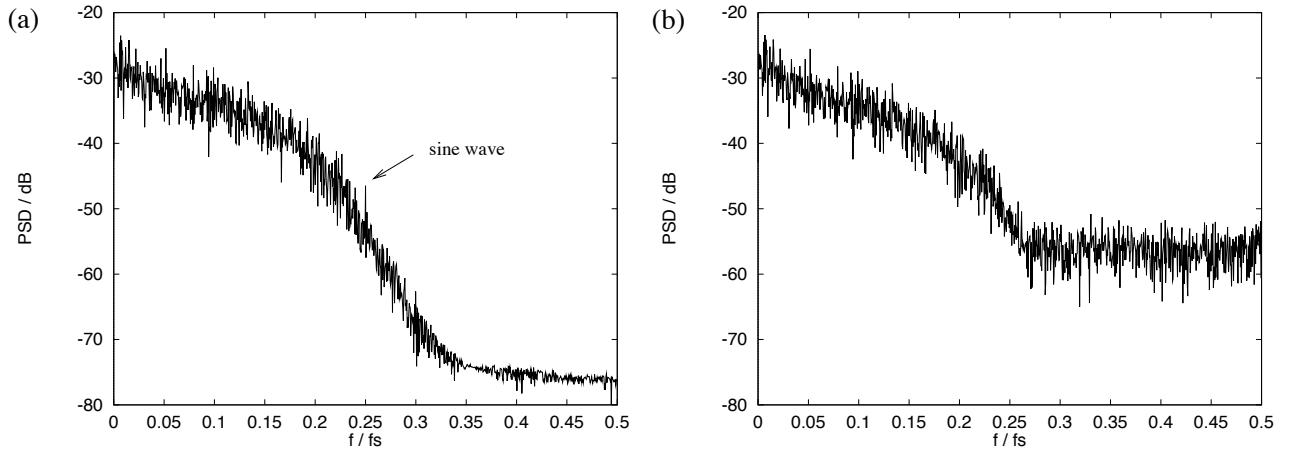


Figure 5.18: PSD : (a) Signal $v(nM)$ (b) Quantised signal $v'(nM)$

The output of the nonlinear model for both schemes is shown in Figure 5.19(a) and (b). The scheme which uses the quantiser has successfully masked the sine wave. Therefore, the output $y(nM)$ does not contain the signal of interest. Without the quantiser the nonlinear model attempts to estimate the signal of interest. The sine wave can be clearly seen in Figure 5.19(a) on the frequency bin $f/f_s = 0.25$.

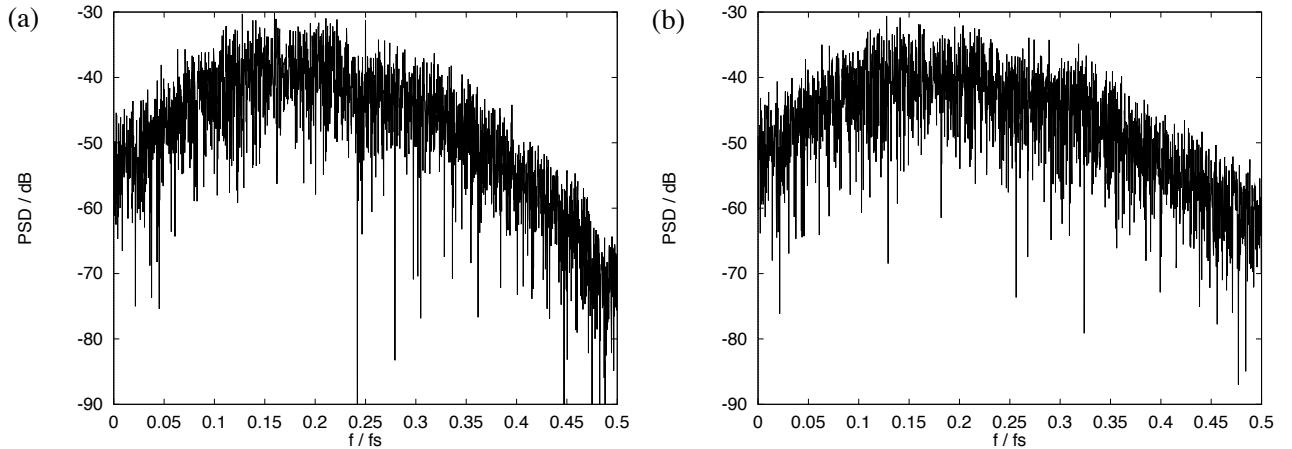


Figure 5.19: PSD : (a) Signal $y(nM)$ without quantiser (b) Signal $y(nM)$ with quantiser

A NRBF network has been trained with 6000 samples. An embedding dimension of 5 was chosen. The number of kernels was 81 and their bandwidth was 0.9. Testing and validating the NRBF network was carried out using further 6000 samples.

The NMSE in dB are depicted in Figure 5.20(a) and (b). The quantisation causes the NMSE to be worse by 13 dB. Although, this might appear a negative result at first sight, it also means that the nonlinear model, without quantisation, fits the signal of interest and, therefore, causes the error to be smaller.

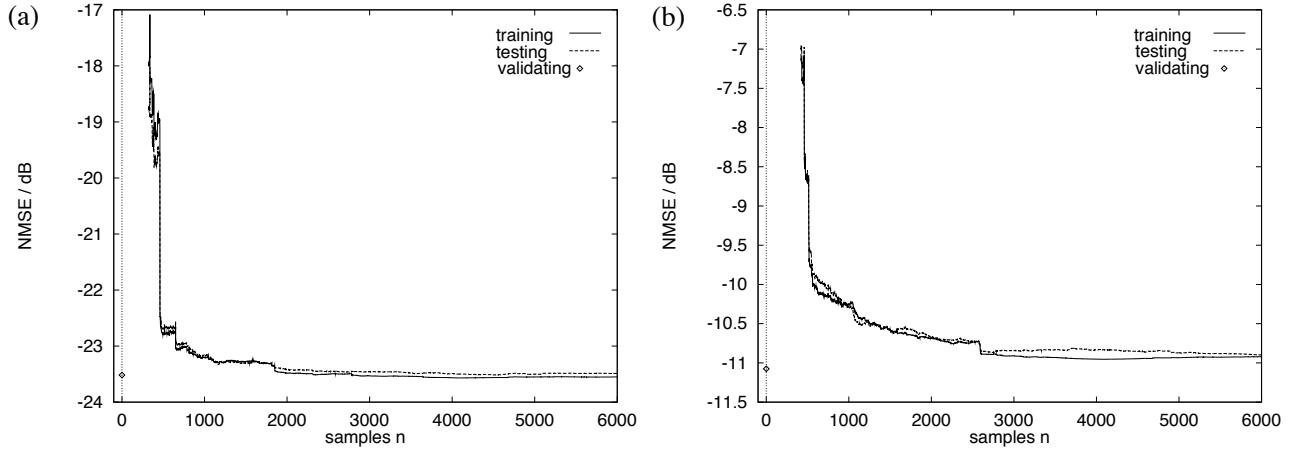


Figure 5.20: NMSE in dB for training, testing and validating : (a) Without quantiser (b) With quantiser

After bandpass filtering and interpolation the recovered signal of interest $\hat{a}(n)$ is obtained. The PSD of the recovered signal of interest, without and with quantisation, can be seen in Figure 5.21(a) and (b), respectively. The energy of the sine wave, without the quantiser, is reduced by about 7 dB. The problem with the scheme with the quantiser is that the quantisation noise causes an additional noise source on the signal of interest.

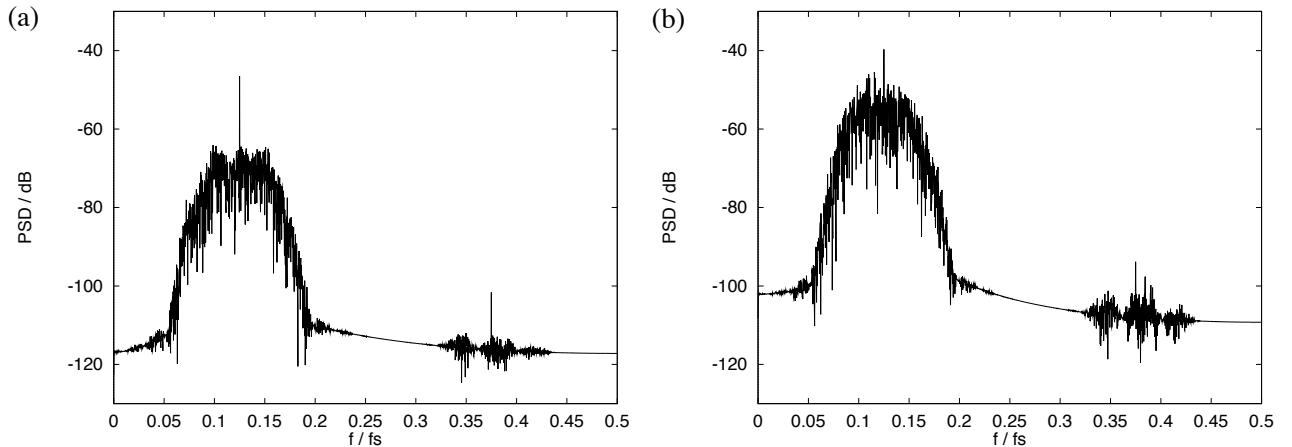


Figure 5.21: PSD : (a) Recovered signal of interest $\hat{a}(n)$ without quantisation (b) Recovered signal of interest $\hat{a}(n)$ with quantisation

The achieved output SNR without quantisation is 1.3 dB and with quantisation -7.5 dB.

Another experiment with the same signals was carried out, where the input SNR is -36 dB. The PSD of the corrupted sine wave is shown in Figure 5.22 from which it can be seen that the sine wave cannot be identified.

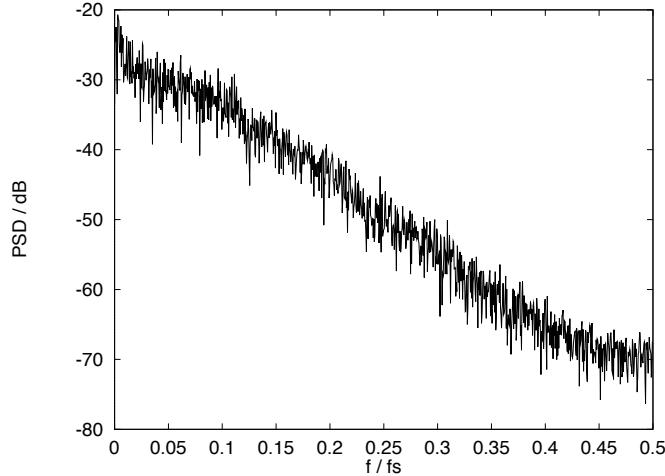


Figure 5.22: PSD : Sine wave in chaotic noise

Even without using a quantiser the output of the nonlinear model does not contain the signal of interest. The scheme is shown in Figure 5.23.

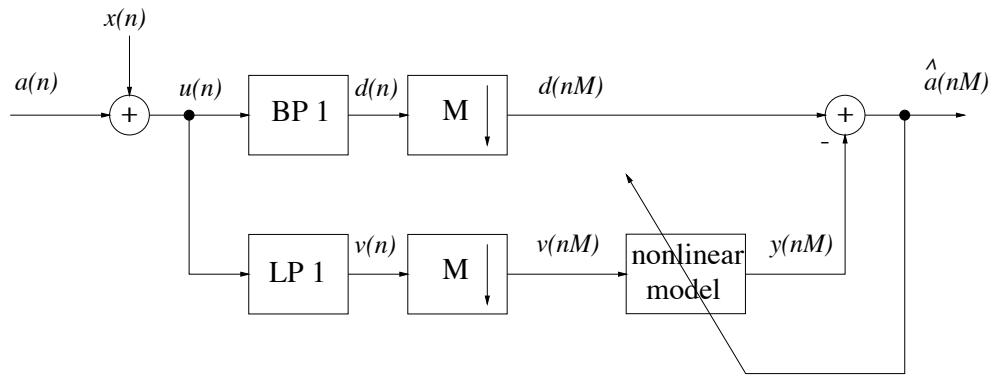


Figure 5.23: Nonlinear interference cancellation

The output of the nonlinear model is shown in Figure 5.24(a) and the estimated signal of interest is shown in Figure 5.24(b).

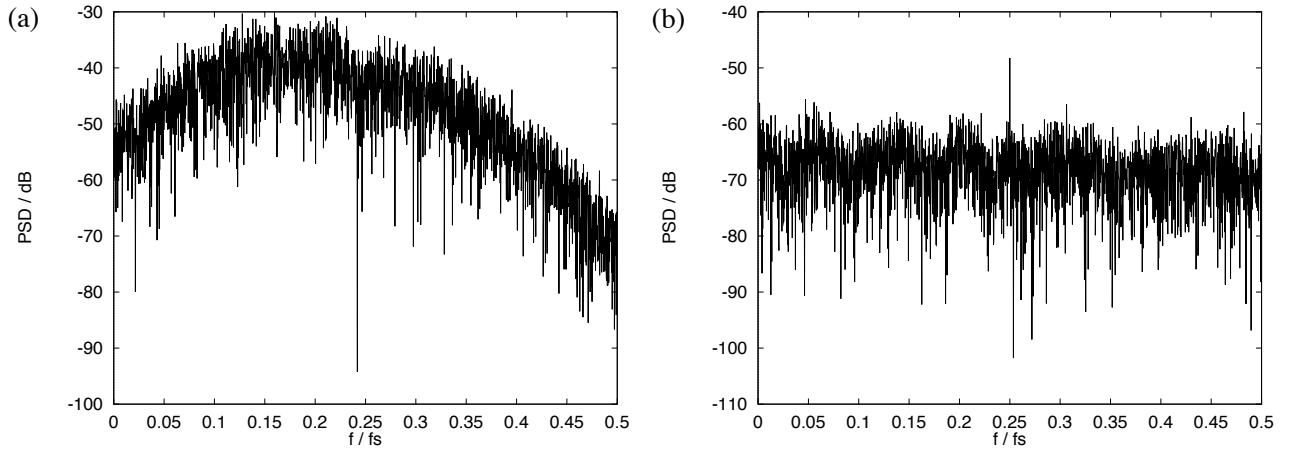


Figure 5.24: PSD : (a) Signal $y(nM)$ (b) Recovered signal of interest $a'(nM)$

The achieved output SNR is -2.6 dB, while linear bandpass filtering techniques such as that shown in Figure 5.8 can only achieve an output SNR of -27 dB.

5.3.3 Discussion

Although the quantisation scheme is able to mask the signal of interest, the method fails. The problem is the amount of quantisation noise that has to be added to the attenuated signal of interest. If a great amount of the signal of interest leaks through the FIR filter a lot of quantisation noise is needed to mask the signal of interest. This causes problems for the nonlinear model which has to find a smooth nonlinear map between the reference and desired signals.

One interesting point could be gathered from the experiments. If the signal of interest is significantly smaller than the nonlinear deterministic noise a noticeable improvement in the output SNR may be achieved. The reason for this is that the energy of the signal of interest in the reference signal is so small that the nonlinear model is not able to detect any correlation and, therefore, only models the nonlinear deterministic noise. This has also been reported in [136], but with further restrictions on the signals. One method requires the signals to be slow time varying [121], the other one assumes that the attractor of chaotic system is hyperbolic [137]. Although the non-orthogonality of the FIR filters is not required the method described here can still detect or even recover narrowbanded signals of interest in heavy chaotic noise.

5.4 Cascaded Normalised Radial Basis Function Networks

In this section cascaded orthogonal FIR filters and NRBF networks are implemented. Their task is to recover three sine waves in a narrow band which is corrupted by broadband noise. The broadband noise is assumed to be nonlinear and deterministic. Section 5.4.1 explains the scheme and in Section 5.4.2 simulations have been carried out. Section 5.4.3 concludes this section with a discussion.

5.4.1 Description of the Scheme

Section 5.2 showed that the lengths of the FIR filters affected the performance of the nonlinear model. The longer the filter the more the chaotic noise was distorted and the more difficult it became for the nonlinear model to estimate the nonlinear inverse of the linear FIR filter. On the other hand, high order FIR filters are necessary when the signal of interest is more complex and the orthogonality of the FIR filters to the signal of interest is not possible. It is well known that high order FIR filters can be broken up into cascaded smaller FIR filters. The idea is to take sequential sets of a very short FIR filter (3 coefficients) and a nonlinear model, which has to estimate the nonlinear inverse of the very short linear FIR filter. To verify this scheme following scenario was set up. Three sine waves in a narrow band with different amplitudes are corrupted by chaotic noise (Lorenz). The task of the cascaded FIR filters and NRBF networks is to recover the sine waves. For simplicity reasons, the FIR filters were designed to be orthogonal to one of the sine waves. A block diagram of the scheme is shown in Figure 5.25.

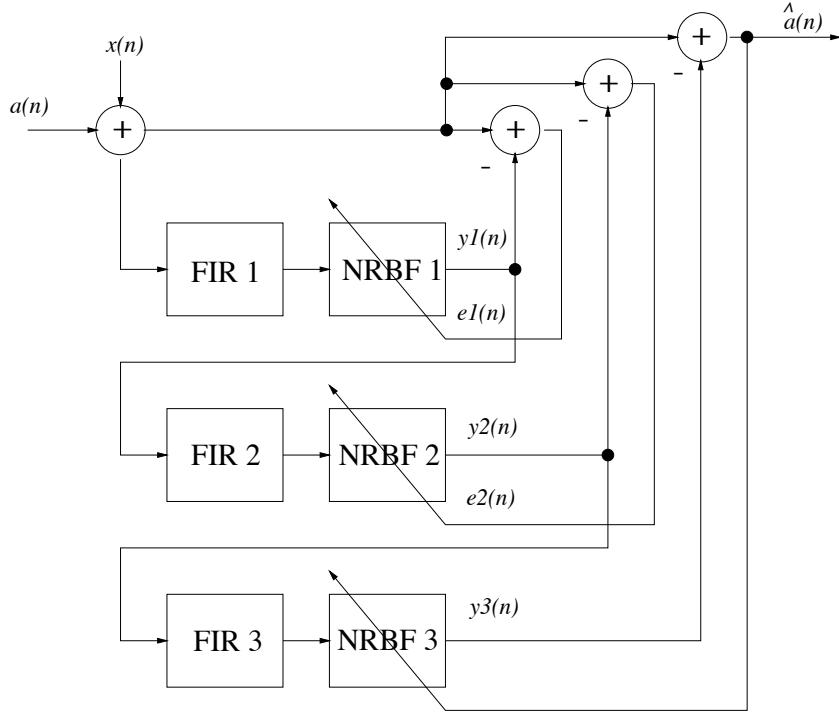


Figure 5.25: Cascaded NRBF networks for nonlinear interference cancellation

5.4.2 Simulations and Results

Three sine waves on the frequency bins $f/fs = 0.1$, $f/fs = 0.125$ and $f/fs = 0.1666$, with amplitudes of 0.5, 0.3 and 0.4, respectively, are corrupted by chaotic noise from the Lorenz system. The parameters of the Lorenz system are the same as the ones in the previous sections. The amplitude of the chaotic noise was chosen to correspond to an input SNR of 8 dB. The PSD of the three sine waves corrupted by the chaotic noise can be seen in Figure 5.26.

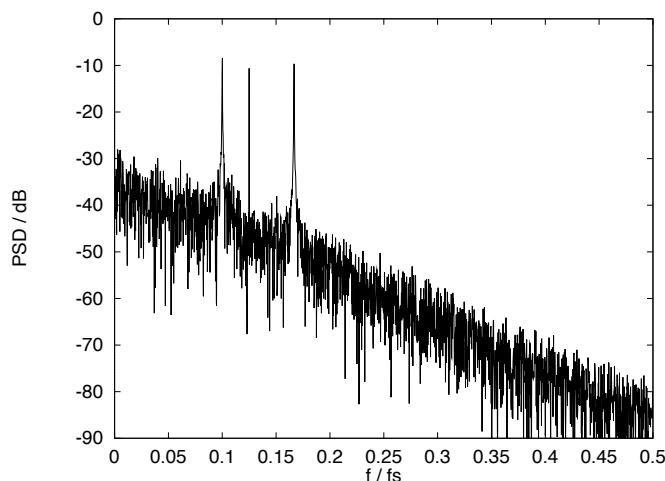


Figure 5.26: PSD : Three sine waves in chaotic noise

In Figure 5.27(a) the PSD of the output $y1(n)$ of the first NRBF network is shown. The first sine wave $f/f_s = 0.1$ has been removed and the chaotic noise restored. The signal $y1(n)$ is fed to the other set, in order to remove the second sine wave at $f/f_s = 0.125$. The PSD of the signal $e1(n)$ is shown in Figure 5.27(b).

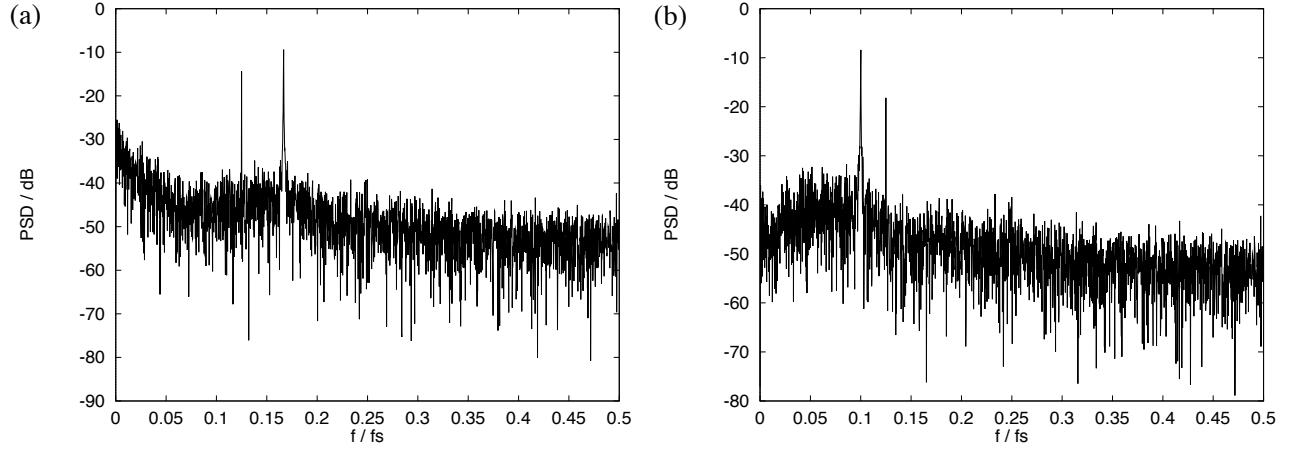


Figure 5.27: PSD : (a) Signal $y1(n)$ (b) Signal $e1(n)$

The PSD of the second output $y2(n)$ is shown in Figure 5.28(a). The second sine wave is almost removed and the distorted chaotic noise caused by the notch filter has been recovered by the NRBF network. The PSD of the signal $e2(n)$ is shown in Figure 5.28(b).

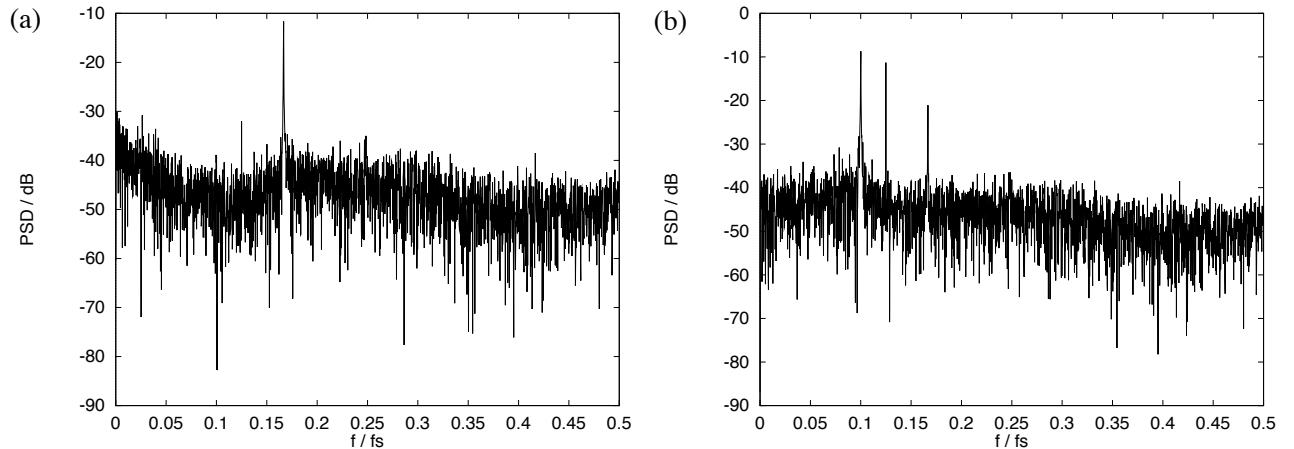


Figure 5.28: PSD : (a) Signal $y2(n)$ (b) Signal $e2(n)$

The last FIR filter removes the sine wave at $f/f_s = 0.1666$. The PSD of the output $y3(n)$ of the NRBF network is shown in Figure 5.29. Ideally, the PSD of $y3(n)$ should not contain the signal of interest. However, the 3 sine waves can be clearly identified. Therefore, the energy of the sine wave in the recovered signal $\hat{a}(n)$ is reduced by around 2 dB each.

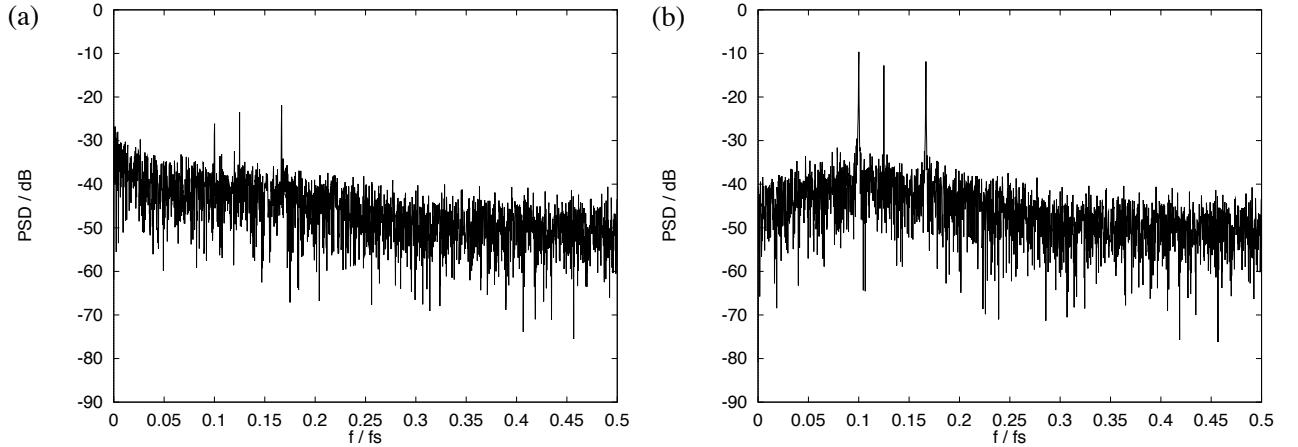


Figure 5.29: PSD : (a) Signal $y3(n)$ (b) Signal $\hat{a}(n)$

It is noticeable, compared to the input scenario in Figure 5.26, that the energy of the chaotic noise has not been significantly reduced in the bandwidth of interest. No noise reduction over the whole bandwidth has been achieved.

5.4.3 Discussion

In this section cascaded sets of FIR filters and NRBF networks were used to recover 3 sine waves from chaotic noise derived from the Lorenz system. Although orthogonal FIR filters, which cancelled the signal of interest completely were implemented, the method failed. This is because the first NRBF network introduces an additional amount of estimation noise, which causes further estimation problems for the second NRBF network and so on. Another drawback, although using orthogonal FIR filters, is that the third NRBF network is able to model the signal of interest. This also degrades the performance of this scheme.

5.5 Chapter Summary

In this chapter three different alternatives to the filter method of the previous chapter were presented. The first one is the new filter method. It exploited additional signal processing techniques, in order to avoid the restrictions from the filter method of the previous chapter. It could be shown that 2 dB improvement in the output SNR was possible compared to conventional linear techniques. It also showed that there is a tradeoff between low and high order FIR filters.

The second method used a quantiser in order to mask the attenuated signal of interest. The quantiser successfully masked the signal of interest, hence, the NRBF network was not able to model the signal of interest.

However, the amount of quantisation noise necessary to mask the attenuated signal of interest caused an additional estimation problem for the NRBF network. Hence, the output of the NRBF network is rather noisy and, therefore, degrades the performance of this scheme. If the signal of interest is significantly smaller than the chaotic noise, it is possible to use the scheme without a quantiser. That is because the NRBF network is not able to detect any correlations between the attenuated signal of interest in the reference signal $v(nM)$ and the desired signal $d(nM)$. A significant improvement in the output SNR could be achieved compared to conventional linear techniques.

The third method exploited the fact that high order FIR filters can be broken up into sequential sets of low order FIR filters. The idea was to estimate the nonlinear inverse of the low order FIR filter in sequence. To make the task easier in the simulation for the NRBF networks, orthogonal FIR filters were implemented. However, the estimation errors of NRBF networks propagated through the cascaded system and caused the output of the third NRBF network to be rather noisy. Hence, this method is not suitable for recovering a narrowband signal of interest from nonlinear deterministic noise.

Chapter 6

Nonlinear Active Noise Control

6.1 Introduction

Active noise control [138] has received a lot of attention in recent years, due to advances in digital signal processing (DSP) and electronics. This technique has been successfully applied to many areas, including heating, ventilating and air conditioning (HVAC) systems [139], exhaust and motor noise [140], headsets [141] and in passenger cabins in aeroplanes [142]. ANC, in general, is based on the principle of the destructive interference between a primary noise source and a secondary source, whose acoustic output is governed by a controller. ANC is an intelligent method which exploits the long wavelengths of low frequency acoustic noise. Using passive sound absorbers to suppress low frequencies is not very useful, because the wavelength of a sound wave of frequency 100 Hz, for example, will have a wavelength of about 3.4 meters in air under normal conditions. This means that passive solutions for low frequencies are expensive in terms of weight and bulk.

The basic idea of active noise control is to generate sound fields which act as destructive interference between the original (primary) sound source and a second sound source. The acoustic output of the second sound source can be controlled. The second sound source, for example, a loudspeaker has to produce an acoustic wave that is exactly out of phase with the acoustic wave produced by the primary source.

To illustrate the physical limitation [143] of only being able to actively control low frequency noise ($< 1 \text{ kHz}$) consider the following example shown in Figure 6.1(a) and (b). Two loudspeakers represented by a square and a cross send out the same sound waves, but in antiphase. In Figure 6.1(a) the sound fields generated by the two loudspeakers, when operating at low frequencies under free field (anechoic) conditions, are illustrated. The peak of one sound wave almost coincides with a trough of the other sound wave and, therefore, destructive interference is achieved globally. If the exciting frequency reaches a certain level, so that the wavelength of the sound waves become comparable with the separation distance of the sources, then destructive (D) and constructive (C) interference occurs. This situation is shown in Figure 6.1(b).

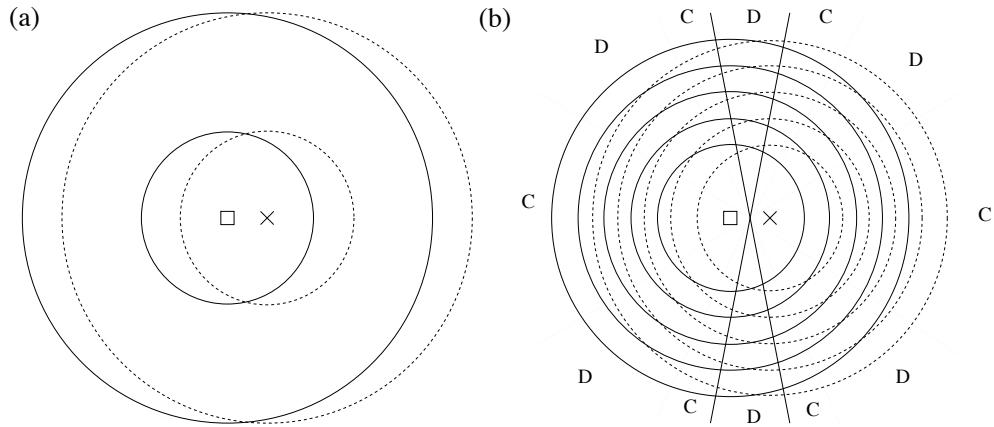


Figure 6.1: Sound waves from out of phase operating acoustic sources (a) Low frequency - global destructive interference (b) High frequency - localised destructive (D) and constructive (C) interference

The localised constructive interference at high frequencies limits the broader application of ANC systems. According to [143] there are three different acoustic objectives to achieve destructive interference.

- Power output minimisation - minimising the output power of primary and secondary sources
- Quiet zone - creating a quiet limited spatial zone around a single microphone
- Power absorption - arranging a phase relationship between loudspeaker and microphone, so that the loudspeaker absorbs power

There are two possible control strategies.

- Feedback
- Feedforward

A generalised block diagram of a feedback control strategy [144, 145] is shown in Figure 6.2. The signal $e(t)$ is measured by the monitor microphone and is fed back to the controller. The controller minimises the power of the signal $e(t)$ by producing an output signal $y(t)$ to the secondary source. The controller not only has to produce a phase shift, but also has to incorporate the transfer function of the secondary path (electroacoustic response of the loudspeaker, acoustic path between speaker and microphone, and the microphone's electroacoustic response). The secondary path has a frequency response which is neither flat nor free of phase shift. This means that the control system can become unstable. It is possible to introduce compensating filters (lead-lag networks) to correct the phase shift [143].

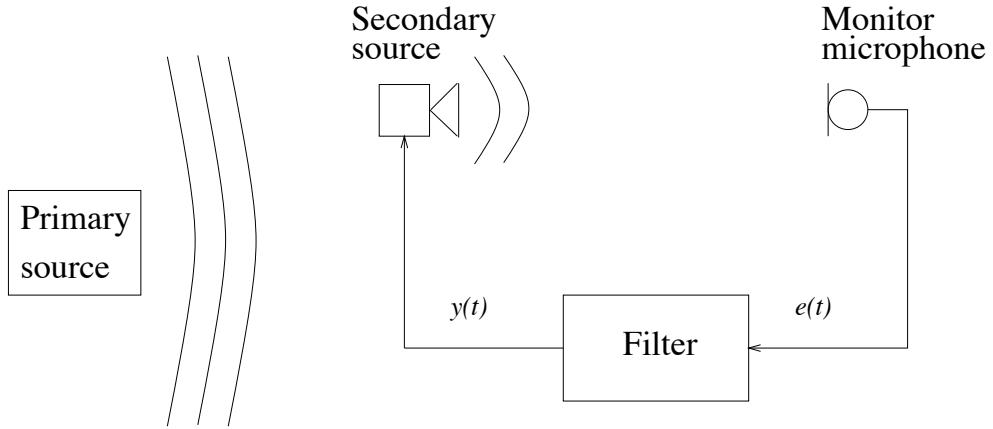


Figure 6.2: Active noise system using feedback control

In Figure 6.3 a feedforward control [146–148] scheme is depicted. In this scheme a reference signal $x(t)$ which is correlated to the primary disturbance is fed into the controller. The controller passes the signal $y(t)$ to a secondary source in order to minimise the power of the signal $e(t)$ at the monitor microphone.

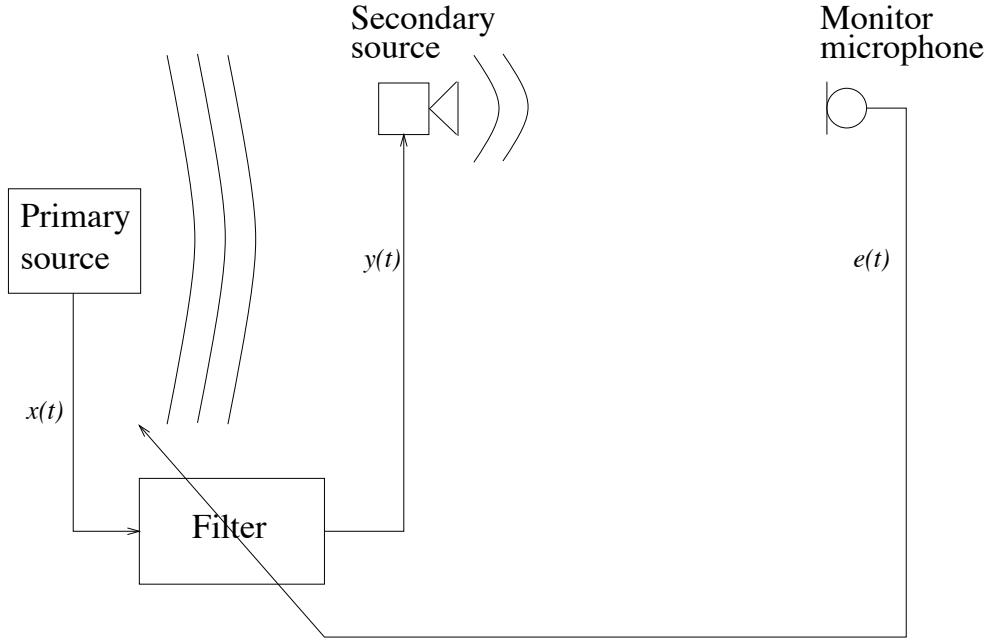


Figure 6.3: Active noise system using feedforward control

The attenuation achievable by ANC systems is also dependent on the location of the secondary source. One of the methods to determine the optimum secondary source location is presented in [149]. The following work carried out in this chapter focuses on a feedforward control strategy in a duct. The acoustic objective is to minimise the output power in the duct using a single detect microphone and an error microphone, and a single loudspeaker. The concepts developed for a single input - single output (SISO) system are usually readily extended to multiple input - multiple output (MIMO) [150–153] systems.

In this chapter a nonlinear active noise control (ANC) technique is devised. Up to now, especially in practise, only conventional linear techniques in active noise control have been used. It is shown that the traditional filtered-x control scheme with a nonlinear controller cannot be used and a new control scheme is devised. Also, a new combined linear and nonlinear controller is developed, which shows very good performance compared to conventional linear techniques.

Section 6.2 describes the background and outlines the current state of the art in ANC. In Section 6.3 a combined linear and nonlinear controller is developed and a new control scheme for ANC in ducts is derived. Simulations and results are presented in Section 6.4 and Section 6.5 concludes this chapter.

6.2 Background

6.2.1 Active Noise Control in Ducts

A typical ANC system in a duct [154,155] is shown in Figure 6.4 and its equivalent block diagram in Figure 6.5. The sound wave in a duct can be regarded as a one-dimensional propagating wave [156], as long its frequencies are below the natural frequency of the first cross sectional mode in the duct.

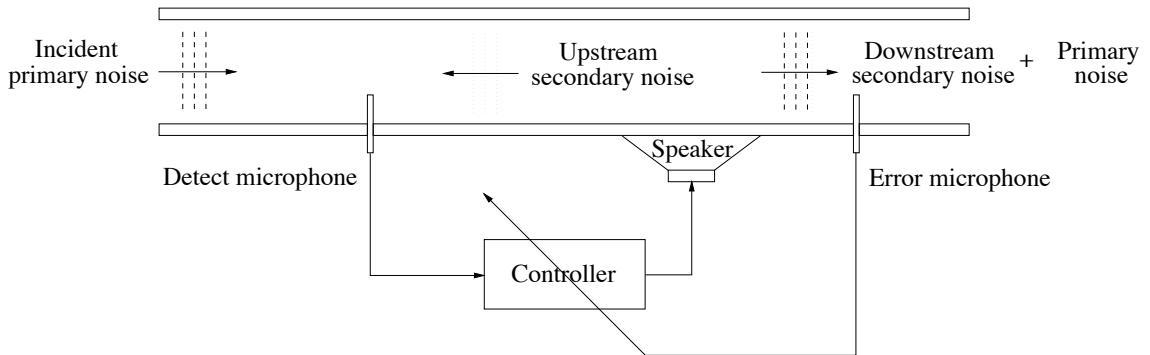


Figure 6.4: Feedforward control in a duct

ANC systems in ducts are usually based on a feedforward control strategy [157–160]. The noise in Figure 6.4 from the primary source travels from left to right, as plane waves [161,162] through the duct. A microphone, located upstream from the secondary source, detects the incident noise waves and supplies the controller with an input signal. The controller sends a signal to the secondary source (i.e. loudspeaker) which is in antiphase with the disturbance. A microphone, located downstream, picks up the residuals and supplies the controller with an error signal. The controller has to be adaptive, because of changes in the environment, degradation of system components (e.g. loudspeaker) and alterations of the noise source.

In some cases a reference signal from the primary noise source can be measured. The advantage is that the

reference signal is not affected by the secondary source. Periodic tonal noise disturbances are usually caused by rotating machines. A well correlated reference signal may be obtained via a mechanical, electrical or optical transducer, which has the same fundamental frequency as the sound emitted [143, 163, 164].

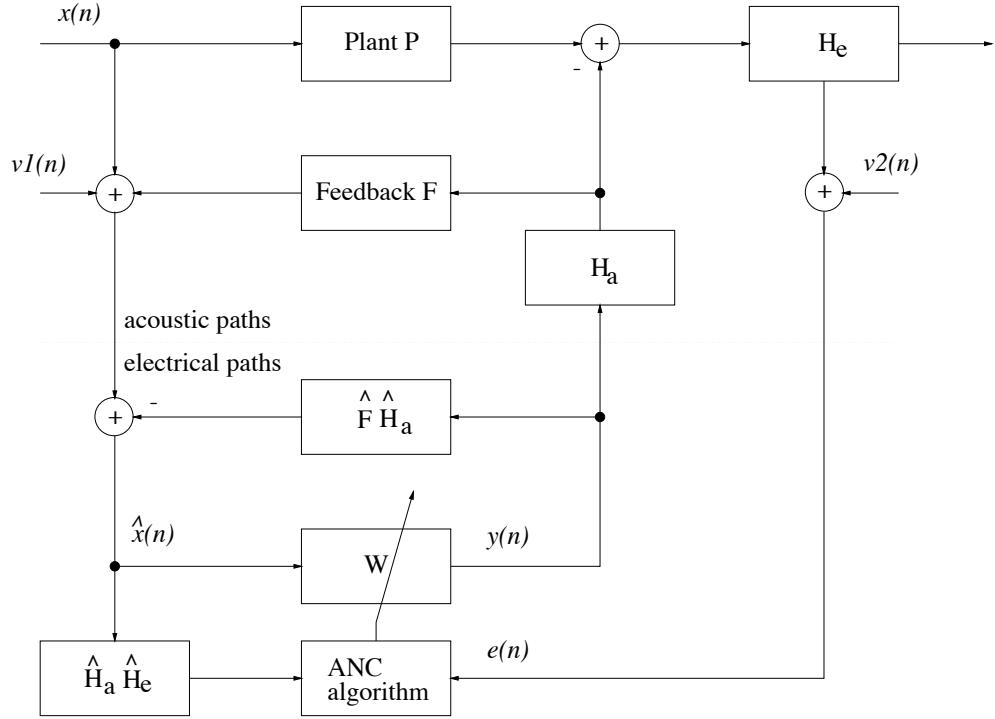


Figure 6.5: Block diagram of the ANC system with conventional adaptive FIR filter and feedback cancellation

The linear adaptive FIR controller W in Figure 6.5 is usually trained by the filtered-x LMS algorithm. $P(z)$ is the transfer function of the duct plant between the detection microphone and the control source. $H_a(z)$ is the transfer function of the actuator, which also represents a driving unit and a reconstruction filter for the D/A conversion. The acoustic path between the control source and error microphone, and the error microphone itself is given as the transfer function $H_e(z)$. The acoustic feedback from the loudspeaker to the detection microphone is represented by the transfer function $F(z)$. To combat the feedback path an adaptive filter with a transfer function $\hat{F}\hat{H}_a$ is modelled off-line and is implemented as shown in Figure 6.5 [143]. The turbulence in the duct causes an additional noise source on the microphones. These noise sources are given as $v1(n)$ and $v2(n)$.

6.2.2 The Filtered-x Algorithm

The variance of the error signal $e(n)$ in Figure 6.5 is usually minimised by a variant of an adaptive least means squares (LMS) algorithm in the controller. As the error signal $e(n)$ is delayed and filtered by the transfer function $H_a(z)H_e(z)$ the standard LMS algorithm cannot be applied to this scheme. The filtered-x LMS [138] algorithm is widely used and requires a transfer function model $\hat{H}_a(z)\hat{H}_e(z)$. It has been shown to be a robust and reliable algorithm for linear ANC systems in [165–169]. The algorithm was first proposed by Morgan [170] in 1980. Widrow *et al.* [171] implemented it in a feedforward control scheme and Burgess [172] in a ANC system, both in 1981. A filtered-x least mean fourth algorithm was proposed in [173].

The controller $W(z)$ has to model the transfer function $P(z)H_a^{-1}(z)$. The error path transfer function $H_e(z)$ does not affect the model $W(z)$ itself, but it does destabilise the adaption process of the LMS algorithm. Therefore, it is necessary to build in an additional model of the transfer function $H_a(z)H_e(z)$ for the algorithm into the input path $x(n)$. The derivation of the filtered-x algorithm is shown in Figures 6.6(a) to (c). Shifting $H_e(z)$ back over the summation point results in Figure 6.6(b). Assuming that the controller $W(z)$ is linear and the adaption process is slow [174], it is possible to shift $H_a(z)H_e(z)$ over the controller $W(z)$. Now it can be seen in Figure 6.6(c), that an additional model of $H_a(z)H_e(z)$ is necessary to filter the input $x(n)$, before it is processed by the LMS algorithm, hence the name 'filtered-x' LMS algorithm.

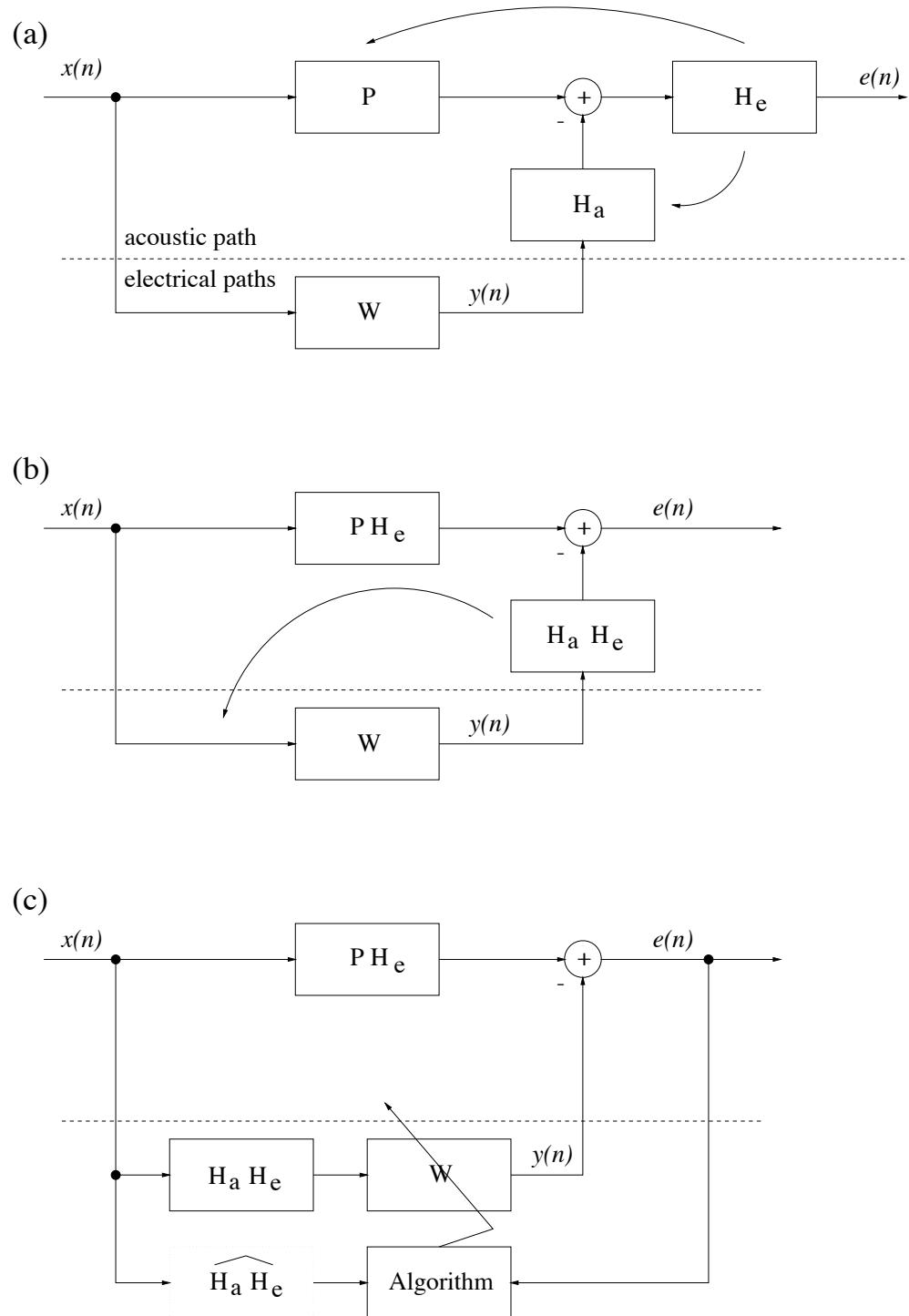


Figure 6.6: Derivation of the filtered-x LMS algorithm : (a) noise canceller; (b) combining transfer functions; (c) adaptive structure

Assuming that the feedback path F has been effectively cancelled, the signal $e(n)$ can be expressed as

$$e(n) = \mathbf{h}_e * \{\mathbf{P} * x(n) - \mathbf{h}_a * (\mathbf{w}_n * x(n))\} \quad (6.1)$$

where $*$ denotes convolution. Assuming that $H_a H_e$ can be represented by a FIR filter of order p

$$\mathbf{h} = [h(0), h(1), \dots, h(p)]^T \quad (6.2)$$

and the controller W as another FIR filter of the order q

$$\mathbf{w}_n = [w_n(0), w_n(1), \dots, w_n(q)]^T \quad (6.3)$$

where the subscript n denotes the time at which the coefficients are updated, then (6.1) may be rewritten as

$$e(n) = d(n) - \sum_{j=0}^p \sum_{i=0}^q h(j) w_{n-j}(i) x(n-j-i) \quad (6.4)$$

where $d(n) = \mathbf{h}_e * (\mathbf{p} * x(n))$. The weights $\{w_n\}$ are updated by the filtered-x LMS algorithm

$$w_{n+1}(i) = w_n(i) + 2\mu e(n) \sum_{j=0}^p \hat{h}(j) x(n-j-i) \quad (6.5)$$

assuming that $w_{n-j}(i) \approx w_n(i)$ the algorithm will converge in the least mean square sense. For this reason there is no benefit to use fast least squares schemes, i.e. recursive least squares (RLS) [175, 176], because the weights $\{w_n\}$ change significantly from one iteration step to the next one. The results of the filtered-x LMS and RLS are displayed in Figures 6.7 and 6.8, respectively.

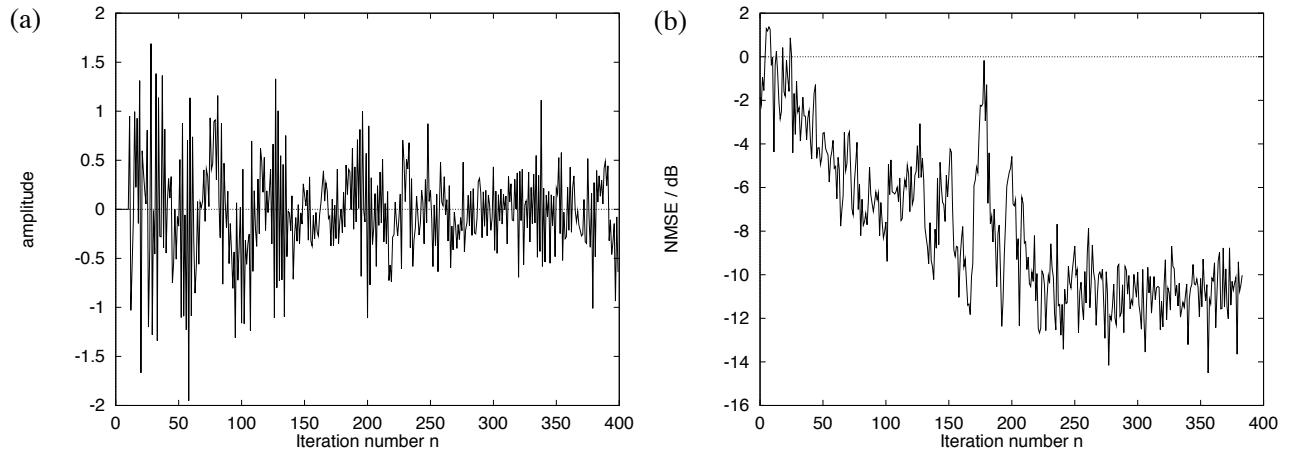


Figure 6.7: Filtered-x LMS, 20 filter coefficients, $\mu = 0.01$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB

The signal $e(n)$ is shown in Figure 6.7(a) and the NMSE in dB in Figure 6.7(b). The adaptive FIR filter was trained by the filtered-x LMS algorithm. The transfer function of H_a is maximum phase and H_e is a delay of 9 samples. This setup is not well suited for a linear FIR controller. Comparing the results achieved with the filtered-x RLS algorithm in Figures 6.8(a) and (b) shows that the error signal $e(n)$ reaches peaks almost 4 times bigger than the error signal without control during training.

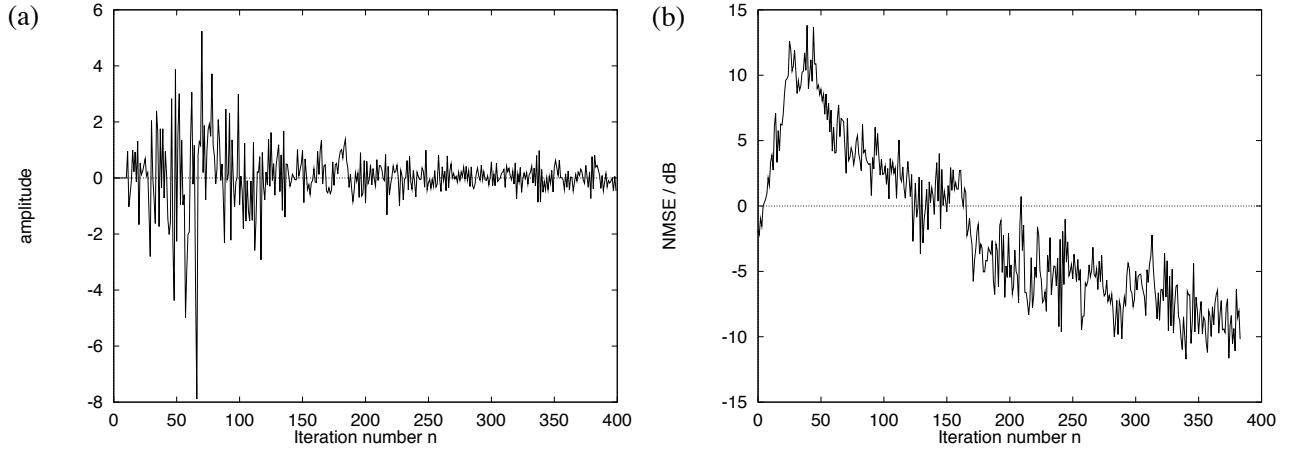


Figure 6.8: Filtered-x RLS, 20 filter coefficients, $\alpha = 1.0$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB

The filtered-x RLS algorithm does not achieve faster convergence than the filtered-x LMS algorithm. In order to exploit fast (in convergence) least squares algorithms other control schemes have to be used. These will be discussed in the next section.

6.2.3 Alternative Linear Control Schemes

In [177] a so-called constrained filtered-x LMS algorithm has been devised. The authors defined another set of residuals

$$\varepsilon(n) = \hat{d}(n) - \sum_{j=0}^p \sum_{i=0}^q \hat{h}(j) w_n(i) x(n-j-i) \quad (6.6)$$

which is similar to (6.4) but without the subscript $-j$ at the weights $\{w_n\}$. The weights $\{w_n\}$ are updated by the constrained filtered-x LMS algorithm

$$w_{n+1}(i) = w_n(i) + 2\mu\varepsilon(n) \sum_{j=0}^p \hat{h}(j) x(n-j-i) \quad (6.7)$$

The residual ε is generated by using only the latest weights $\{w_n\}$. This makes the weights *a priori* stationary. The block diagram is shown in Figure 6.9. It is quite clear that this scheme is rather computationally expensive. Three models for $\hat{H}_a \hat{H}_e$ and $(q+1)$ controllers W have to be implemented.

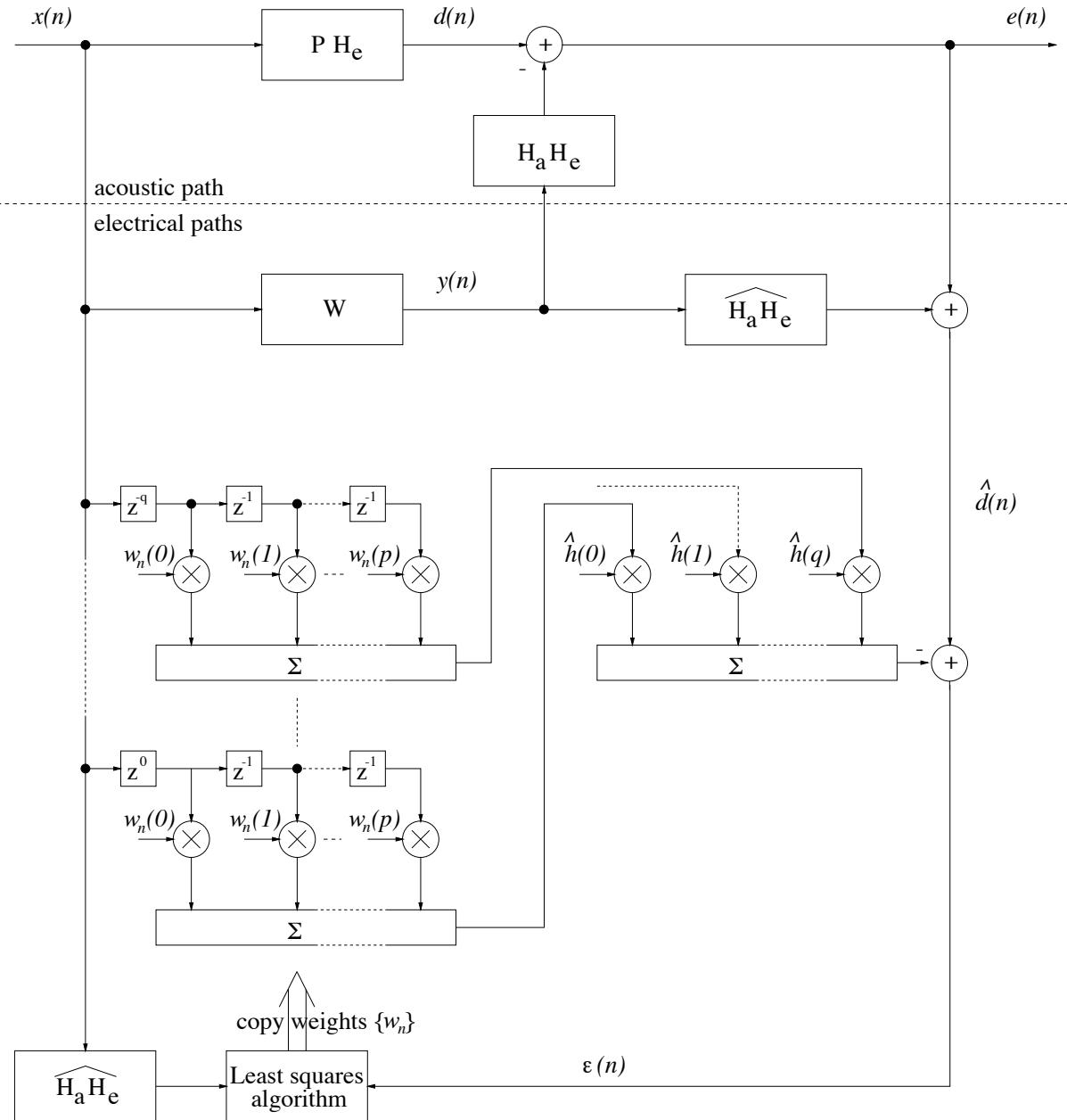


Figure 6.9: Block diagram of the 'constrained filtered-x' LMS algorithm

The same setup as in the previous simulation was again used to compare convergence speed of the LMS and RLS algorithms. The setup is rather ill-conditioned for a linear controller and, therefore, the constrained filtered-x LMS in Figure 6.10 only converges marginally faster than the filtered-x LMS algorithm in Figure 6.7. The NMSE does not oscillate as much in Figure 6.10(b) as in Figure 6.7.

The constrained filtered-x RLS algorithm in Figure 6.11 shows good convergence speed and no high peaks during training in the error signal $e(n)$ compared to Figure 6.8.

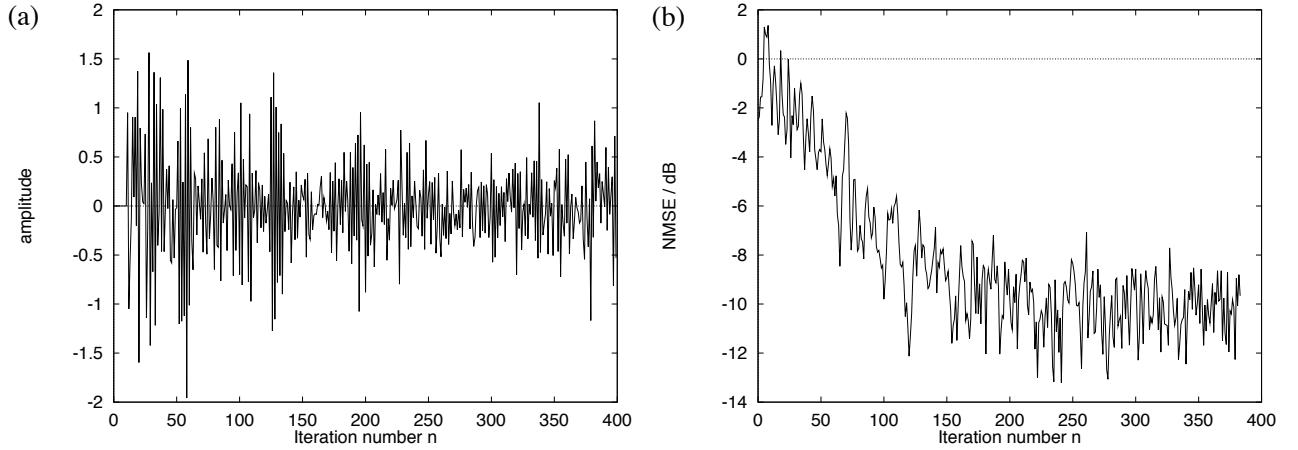


Figure 6.10: Constrained filtered-x LMS, 20 filter coefficients, $\mu = 0.01$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB

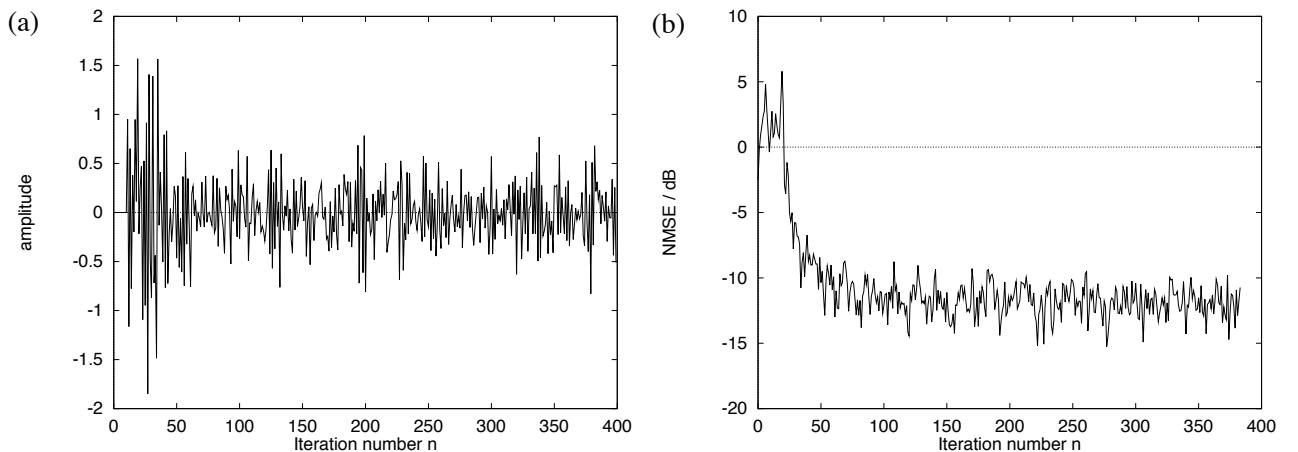


Figure 6.11: Constrained filtered-x RLS, 20 filter coefficients, $\alpha = 1.0$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB

The most widely used control scheme [178, 179] for more sophisticated least squares algorithms, rather than using the filtered-x LMS is shown in Figure 6.12. A fast RLS algorithm [180] and IIR controllers have been implemented in this scheme [181–184].

Again, as in the constrained version, a modified residual is defined

$$e'(n) = \hat{d}(n) - \sum_{j=0}^p \sum_{i=0}^q \hat{h}(j) w_n(i) x(n-j-i) \quad (6.8)$$

The weights $\{w_n\}$ are updated by the modified filtered-x LMS algorithm

$$w_{n+1}(i) = w_n(i) + 2\mu e'(n) \sum_{j=0}^p \hat{h}(j) x(n-j-i) \quad (6.9)$$

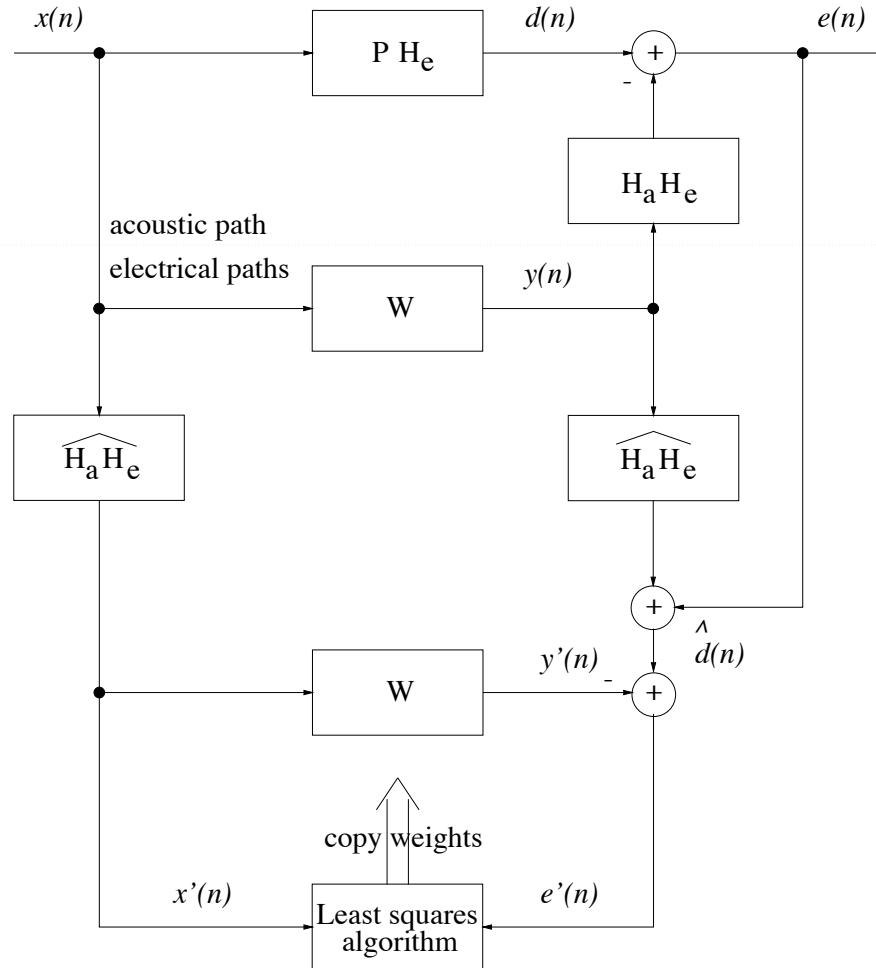


Figure 6.12: Modified filtered-x scheme with an additional 'secondary path' $\hat{H}_a(z)\hat{H}_e(z)$ and controller W

The same setup for the simulations as in the previous ones was used. The modified filtered-x LMS algorithm in Figure 6.13 has similar convergence behaviour as the constrained filtered-x LMS in Figure 6.10.

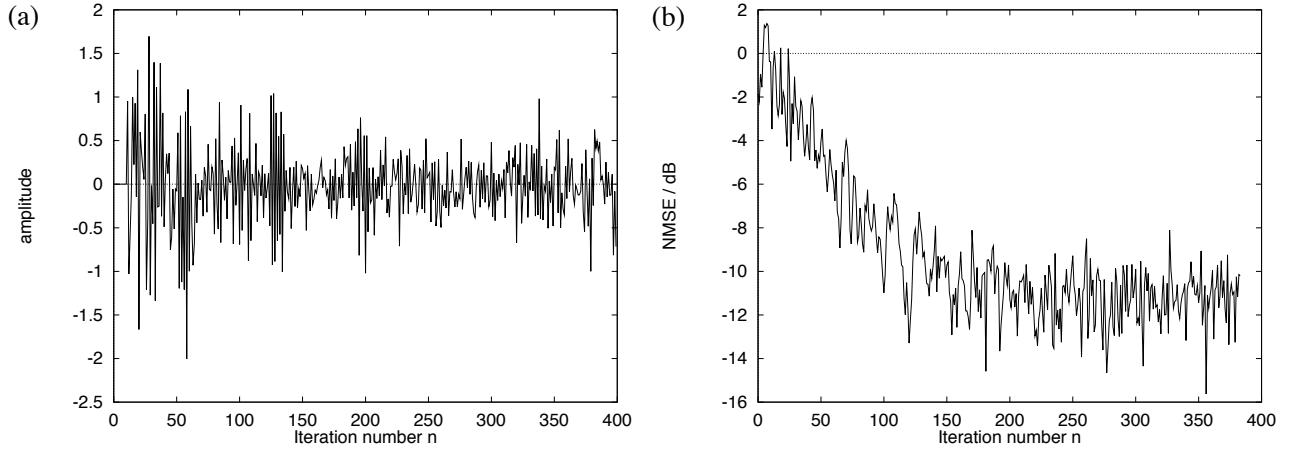


Figure 6.13: Modified filtered-x LMS, 20 filter coefficients, $\mu = 0.01$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB

The modified filtered-x RLS algorithm in Figure 6.14 shows the same rate of convergence as in Figure 6.11. The NMSE in Figure 6.14(b) during training is not as high as in Figure 6.11(b).

In Section 6.3.2, a control scheme with only one additional model $(\hat{H}_a \hat{H}_e)^{-1}$ using a block least squares algorithm is presented.

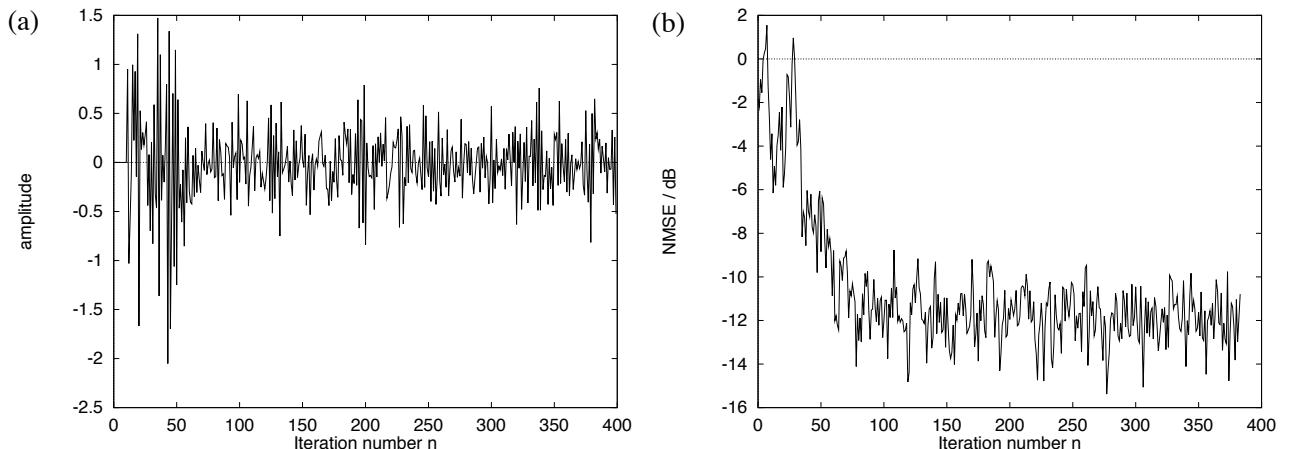


Figure 6.14: Modified filtered-x RLS, 20 filter coefficients, $\alpha = 1.0$, 25 ensembles, $x(n) \sim N(0, 1.0)$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e(z) = \hat{H}_e(z) = z^{-9}$, $P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$, $F(z) = 0$, $v1(n) = v2(n) = 0$: (a) $e(n)$ (b) NMSE / dB

6.3 Nonlinear Active Noise Control in a Duct

6.3.1 Introduction

There are several situations where a nonlinear controller may perform better than conventional linear techniques. First of all the components of the ANC system might exhibit nonlinearities themselves. The loudspeaker, for instance, can excite not only the frequency of interest but also its associated harmonics [185]. This scenario has been investigated by Snyder *et al.* in [186] and by Wangler *et al.* in [79]. In [186] a neural network was implemented as a controller and a second one for modelling the cancellation path transfer function. The authors in [79] used a polynomial filter and a neural network, which were trained by genetic algorithms. In [187] a dynamic recurrent neural network was used as a controller to model the antiphase noise and to combat the effects of the feedback path, simultaneously. A neural network was implemented in [19] to reduce the ANC system's physical size in a duct. The noise from the fan in the duct was shown to be chaotic and therefore it was possible to predict the chaotic noise in the short term. Impulsive noise modelled by an alpha stable process [188] was considered as the primary noise source in [189].

An alternative motivation for the use of a nonlinear controller is provided when the transfer function $H_a(z)$ is nonminimum phase [165]. The FIR controller W in Figure 6.5 has to identify the cascaded systems $P(z)$ and $H_a^{-1}(z)$. Both systems are assumed to be linear. If $H_a(z)$ is minimum phase its inverse [190] is a stable IIR filter. To approximate an IIR filter a high order FIR filter may be designed. In the case that $H_a(z)$ is nonminimum phase [165], the inverse demands a non-causal linear IIR filter. Non-causal filters are not physical realisable, and therefore, are only poorly approximated by FIR filters. However, if the input signal $x[n]$ is stochastic non-Gaussian [101] or nonlinear deterministic [1] a nonlinear inverse to the transfer function $H_a(z)$ exists and may produce an improvement in cancellation performance. This scenario is the main focus of the following sections and will be described in detail.

Constructing a nonlinear inverse filter to a linear filter to cancel nonlinear deterministic noise has been investigated in [1]. Finding an inverse with non-Gaussian stochastic input is also encountered in numerous channel equalisation problems, *e.g.* in [101].

The low-frequency noise in the duct is usually assumed to be broadband random noise or periodic tonal noise [143]. However, today it is known that many of these noise processes arise from nonlinear dynamical systems. Chaotic time series are a typical example of aperiodic time series which appear to be a stochastic process when analysed with second order statistics (see Chapter 2). Deterministic nonlinear behaviour can arise from all kinds off different physical systems as described in Chapter 4.

The filtered-x LMS algorithm is implicitly linear and hence cannot be easily applied to train a nonlinear controller. Therefore other schemes have to be used and one of them is presented in Section 6.3.2. Section 6.3.3 describes how a nonlinear inverse to a nonminimum phase linear filter may exist and derives a mathem-

atical expression for the controller. In Section 6.3.4 various simulations and results are presented. Section 6.3.5 discusses the results obtained by the simulations.

6.3.2 Nonlinear Control of a Linear Plant

It is clear from the simple block shifting procedure in Section 6.2.2 that the filtered-x scheme is not valid for a nonlinear controller. To circumvent this restriction the ANC system was reconfigured to allow a nonlinear controller. The block diagram is shown in Figure 6.15.

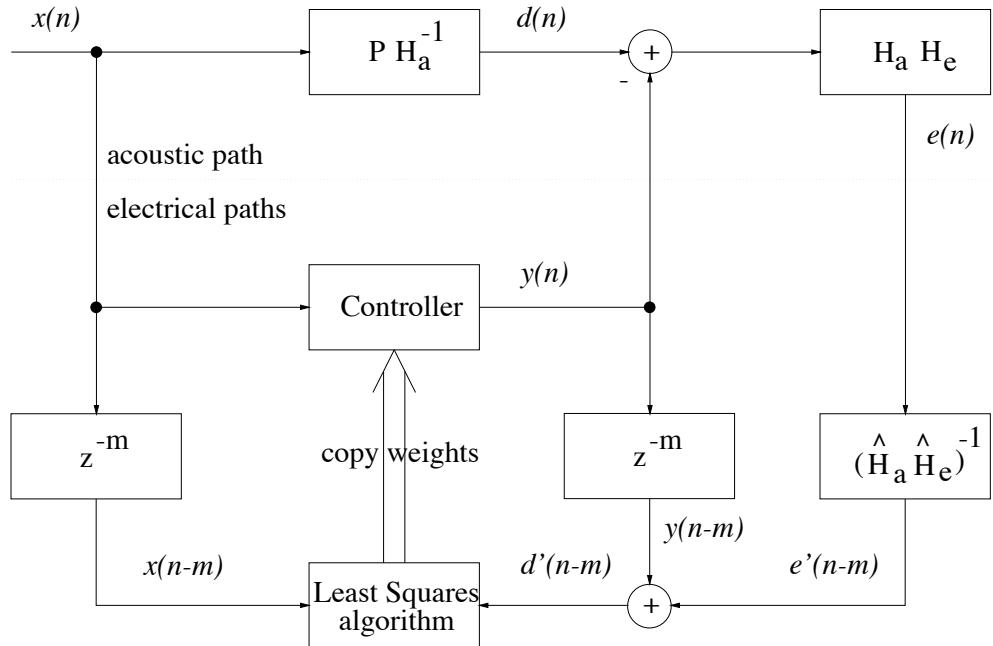


Figure 6.15: Block diagram of the ANC system used in simulations

This control scheme does not need as many models as the constrained or the modified control schemes presented in Section 6.2.3. The scheme uses one additional model $(\hat{H}_a \hat{H}_e)^{-1}$ to estimate a delayed desired signal. Two similar approaches [143] were proposed by Warnaka *et al.* [191] in 1984 and by Poole *et al.* [192] in 1984.

To estimate the actual error signal $e(n)$ it is necessary to build the transfer function $(\hat{H}_a(z)\hat{H}_e(z))^{-1}$ into the error path. The inverse of $H_a(z)H_e(z)$ is modelled off-line, using only the loudspeaker as a white noise source. Fortunately it is possible to use an inverse modelling delay z^{-m} to estimate the inverse more accurately. To compensate for the inverse modelling delay the signals $x(n)$ and $y(n)$ are delayed by the same amount. The signals $y(n - m)$ and $e'(n - m)$ are summed to form the desired signal $d'(n - m)$. The delayed input signal $x(n - m)$ and the delayed desired signal $d'(n - m)$ are supplied to a least squares algorithm.

Another advantage in using this scheme, is that more sophisticated least squares algorithms (*e.g.* Householder transformations [193] (see Appendix A) or singular value decomposition (SVD) [127]) can be implemented, which are especially useful for training a nonlinear controller with a coloured input $x(n)$.

Another possible nonlinear control scheme would be to use a nonlinear controller which consists of two parts. The first part being a nonlinear expansion from the input vector \mathbf{x}_M and the second part a set of linear weights. NRBF networks and VS filter have that kind of structure. Then it would be possible to shift $H_a(z)H_e(z)$ (i.e. in Figure 6.6(b)) only over the second linear part of the controller $W(z)$. This procedure would lead to a nonlinear filtered-x scheme. The nonlinear control scheme mentioned above in this section appears to be more sophisticated and, therefore, was used in the simulations.

6.3.3 Development of a Combined Linear and Nonlinear Controller

If the controller is to model $P(z)H_a^{-1}(z)$ when the actuator transfer function $H_a(z)$ is nonminimum phase [165], a causal linear controller will be sub optimum since it cannot, by definition, characterise the anticausal part of $H_a^{-1}(z)$. The derivation of expressions for the combined controller assumes that the feedback path $F(z)$ has been effectively cancelled by $\hat{F}(z)\hat{H}_a(z)$. This kind of elimination of the feedback path has been reported to be very successful in [180]. The resulting block diagram is shown in Figure 6.16. The transfer function $H_a(z)$ is

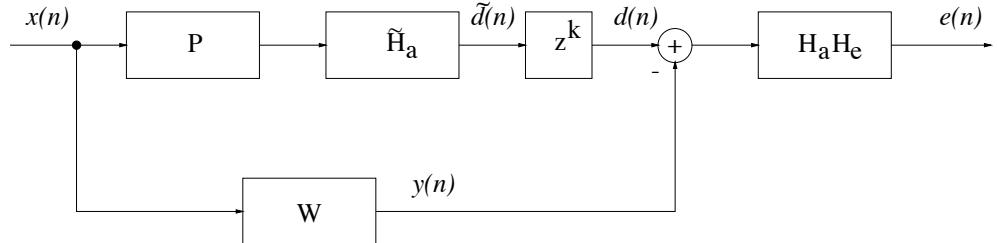


Figure 6.16: Block diagram of ANC system for derivation of the combined controller

assumed to be nonminimum phase with a noncausal inverse $H_a^{-1}(z)$. Define $\tilde{H}_a(z)$ as a causal approximation to the inverse of $H_a(z)$ such that

$$H_a(z)\tilde{H}_a(z) \doteq z^{-k} \quad (6.10)$$

where k is an integer. The quality of approximation improves as the lag k is increased. Thus the transfer function $H_a^{-1}(z)$ can be approximated as

$$H_a^{-1}(z) \doteq z^k \tilde{H}_a(z) \quad (6.11)$$

The function of the controller W in Figure 6.16 is to produce a k -step ahead prediction of the output of the causal linear system $P(z)\tilde{H}_a(z)$ based on the input vector $\mathbf{x}_N(n)$. Thus

$$d(n) = \tilde{d}(n+k) = f(\mathbf{x}_N(n)) \quad (6.12)$$

with

$$\mathbf{x}_N(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T$$

where $f(\cdot)$ is some scalar function, possibly nonlinear, of the vector $\mathbf{x}_N(n)$. Given $\mathbf{x}_N(n)$, the minimum mean square error (MMSE) estimate of $d(n)$ is obtained by the conditional expectation [194], or Bayesian estimate :

$$\hat{d}(n) = E[d(n)|\mathbf{x}_N(n)] \quad (6.13)$$

Assuming that the transfer function $P(z)\tilde{H}_a(z)$ can be approximated by a FIR filter the desired signal is given by

$$d(n) = \sum_{i=0}^{N-1} h_i x(n+k-i) \quad (6.14)$$

where $\{h_i\}$ are the coefficients of the causal transfer function $P(z)\tilde{H}_a(z)$. Equation (6.14) can be split into two parts as

$$\begin{aligned} d(n) &= d_1(n) + d_2(n) \\ &= \sum_{i=0}^{k-1} h_i x(n+k-i) + \sum_{i=k}^{N-1} h_i x(n+k-i) \end{aligned} \quad (6.15)$$

Thus (6.13) can be rewritten as

$$\begin{aligned} \hat{d}(n) &= E[d_1(n) + d_2(n)|\mathbf{x}_N(n)] \\ &= E[d_1(n)|\mathbf{x}_N(n)] + E[d_2(n)|\mathbf{x}_N(n)] \end{aligned} \quad (6.16)$$

The second part of (6.16) is a linear system identification problem

$$\hat{d}_2(n) = E\left[\sum_{i=k}^{N-1} h_i x(n+k-i)|\mathbf{x}_N\right] \quad (6.17)$$

The optimum solution would be

$$\begin{aligned} d_2(n) &= \sum_{i=k}^{N-1} h_i x(n+k-i) \\ &= \sum_{j=0}^{N-1-k} h_{j+k} x(n-j) \\ &= \sum_{j=0}^{N-1-k} g_j x(n-j) \end{aligned} \quad (6.18)$$

Hence, without loss of generality

$$d_2(n) = \mathbf{g}^T \mathbf{x}_N(n) \quad (6.19)$$

where

$$\mathbf{g}^T = [g_0, \dots, g_{N-1}]$$

The first part of (6.16) requires multi-step prediction.

$$\begin{aligned}\hat{d}_1(n) &= E[d_1(n)|\mathbf{x}_N(n)] \\ &= E\left[\sum_{i=0}^{k-1} h_i x(n+k-i)|\mathbf{x}_N(n)\right] \\ &= \sum_{i=0}^{k-1} h_i E[x(n+k-i)|\mathbf{x}_N(n)]\end{aligned}\tag{6.20}$$

Thus $\hat{d}_1(n)$ is a linear combination of multi-step predictors of the input signal $x(n)$. It is usually not necessary to embed $\mathbf{x}_N(n)$ into as many as N dimensions so for modelling, a smaller dimension $M < N$ may be more appropriate.

$$\hat{d}_1(n) = \sum_{i=0}^{k-1} h_i E[x(n+k-i)|\mathbf{x}_M(n)]\tag{6.21}$$

The optimal prediction $E[x(n+k-i)|\mathbf{x}_M(n)]$ may be approximated in practice with many feedforward networks e.g. multilayer perceptrons (MLP), radial basis functions (RBF) or Volterra series (VS). Here, only the latter two will be considered, because of their ease of training. Thus, $x(n+k-i)$ can be approximated by

$$E[x(n+k-i)|\mathbf{x}_M(n)] \hat{=} \sum_{j=1}^L \mu_{ij} \Phi_j(\mathbf{x}_M(n))\tag{6.22}$$

where L is the number of the nonlinear kernels $\{\Phi_j\}$ and linear weights $\{\mu_{ij}\}$. Therefore an estimate of $d_1(n)$ would have the form

$$\hat{d}_1(n) = \sum_{i=0}^{k-1} h_i \sum_{j=1}^L \mu_{ij} \Phi_j(\mathbf{x}_M(n))\tag{6.23}$$

Merging the two linear layers $\{h_i\}$ and $\{\mu_{ij}\}$ together, this becomes

$$\hat{d}_1(n) = \sum_{j=1}^L \omega_j \Phi_j(\mathbf{x}_M(n))\tag{6.24}$$

This removes any requirement to identify the weights $\{h_i\}$ and $\{\mu_{ij}\}$ separately. This also removes any requirement for explicit knowledge of the lag k . Finally, an estimate $y(n)$ of $d(n)$ can be found by substituting (6.19) and (6.24) in (6.16)

$$y(n) = \sum_{j=1}^L \omega_j \Phi_j(\mathbf{x}_M(n)) + \mathbf{g}^T \mathbf{x}_N(n)\tag{6.25}$$

As in [99] this can also be written as a vector inner product

$$y(n) = \mathbf{w}^T \mathbf{z}(n)\tag{6.26}$$

where

$$\begin{aligned}\mathbf{z}^T(n) &= [\Phi_1, \Phi_2, \dots, \Phi_L, x(n), x(n-1), \dots, x(n-N+1)] \\ \mathbf{w}^T &= [\omega_1, \omega_2, \dots, \omega_L, g_0, g_1, \dots, g_{N-1}]\end{aligned}$$

For this linear-in-the-parameter nonlinear architecture, the weight vector \mathbf{w} can be estimated using a least squares algorithm. No explicit knowledge of the lag term k is required in (6.25). To evaluate the linear combination of the nonlinear and linear parts in (6.25) a variety of different controllers are considered in Section 6.3.4.

Based on (6.25) it is possible to infer how the nature of the controller will be affected by the input signal characteristics and the phase response of the actuator transfer function $H_a(z)$. In particular, if $H_a(z)$ is minimum phase, no lag k is required and the inverse is purely causal. Hence a predictive component such as (6.22) is not required and (6.25) reduces to the conventional linear controller [127,195]. If the transfer function is nonminimum phase three conditions can be identified, which are dependent on the nature of the input sequence:

- $x(n)$ is independently identically distributed (i.i.d.) - no prediction is possible and hence poor results are to be expected.
- $x(n)$ is correlated in time and Gaussian in which case (6.22) reduces to a linear predictor and the whole controller of (6.25) is linear.
- $x(n)$ is non-Gaussian and non-i.i.d., in which case (6.25) may produce better results than a purely linear controller.

The linear system identification part in (6.25) requires an input signal $\mathbf{x}_N(n)$ which is broadband. As long as the noise excites all the frequencies necessary to identify the linear system, the linear part does not require any assumptions about the distribution or nonlinear deterministic characteristics in the input signal $\mathbf{x}_N(n)$.

The nonlinear prediction part, though, requires that there is some nonlinear mapping between adjacent input samples. If the noise is nonlinear and deterministic, i.e. a chaotic time series, it is possible to predict the noise in the short term [1]. As the function f in $x(n+k) = f(\mathbf{x}_M(n))$ is generally nonlinear to generate chaos, it is necessary to use a nonlinear model for the k-step prediction task. Chaotic time series which arise from dynamical systems are clearly non-Gaussian but are also deterministic. It is possible to embed the finite dimensional manifold θ of the dynamical system with the methods of delays [1]. As the system evolves through time the measured values of the time series in the tapped delay line describe a trajectory in an embedded state space. This embedded attractor does not occupy the whole state space, in strong contrast to some stochastic processes, which fill out more or less the whole state space.

6.3.4 Simulations and Results

In the following simulations the acoustic feedback path $F(z)$ in the duct is neglected for the two following reasons. Firstly it is assumed that the off-line modelled filter $\hat{F}(z)\hat{H}_a(z)$ eliminates the feedback and secondly that uni-directional microphones are used. The noise caused by turbulence in the duct on the microphones, is assumed to be zero. The resulting block diagram is shown in Figure 6.15.

To justify the necessity of implementing a combined linear and nonlinear controller the following simulations were carried out. The primary noise source was modelled by the Logistic map. The time series $x(n)$ generated by the Logistic map is delta correlated [196]. The actuator $H_a(z)$ was chosen to be maximum phase and the other transfer functions had unit gain. The NMSE in dB can be seen in Figure 6.17. The linear FIR controller with 10 taps is not able to attenuate the noise. The nonlinear controller, a Volterra filter with quadratic and cubic terms, achieves slightly better performance than the combined linear and nonlinear controller. The setup of this simulation provides a purely nonlinear task for the controller and, therefore, it is not surprising that the nonlinear controller achieves the best results.

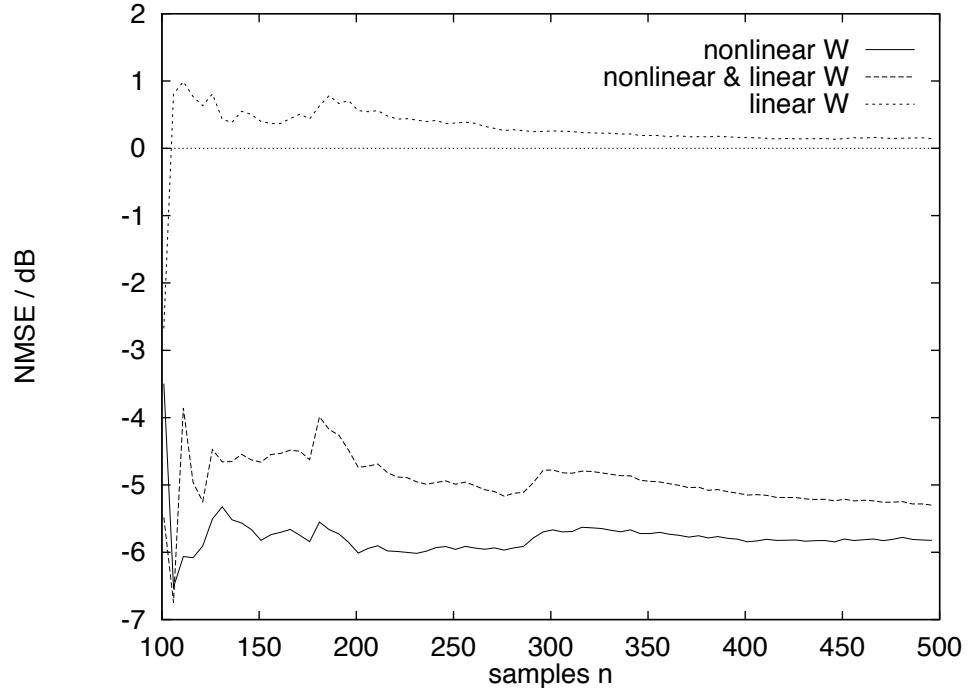


Figure 6.17: NMSE in dB for 3 different controllers : $x(n) = 4x(n-1)(1-x(n-1))$, $M = 3$, $N = 10$, $H_a(z) = \hat{H}_a(z) = 0.5 + z^{-1}$, $H_e = \hat{H}_e = 1$, $P(z) = 1$, $F(z) = 0$, $v1(n) = v2(n) = 0$, $m = 16$, 32 taps for $(\hat{H}_a\hat{H}_e)^{-1}$

The next simulation is a purely linear task for the controller. The same three controllers were used, but this time $H_a(z)$ is minimum phase. The NMSE in dB can be seen in Figure 6.18.

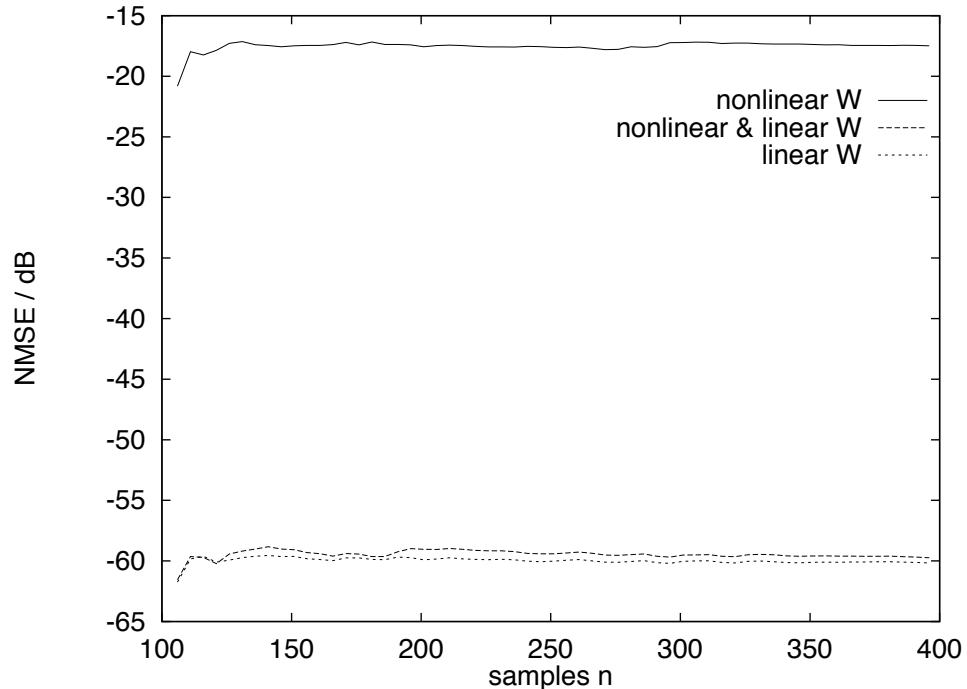


Figure 6.18: NMSE in dB for 3 different controllers : $x(n) = 4x(n-1)(1-x(n-1))$, $M = 3$, $N = 10$, $H_a(z) = \hat{H}_a(z) = 1 + 0.5z^{-1}$, $H_e = \hat{H}_e = 1$, $P(z) = 1$, $F(z) = 0$, $v1(n) = v2(n) = 0$, $m = 0$, 32 taps for $(\hat{H}_a \hat{H}_e)^{-1}$

The linear FIR controller and the combined one achieve the best result. From those two simulations it can be seen that the combined linear and nonlinear controller is suitable for a broad range of control problems.

The next simulations were carried out to compare different controllers with different primary noise sources. Seven different input noise signals $x(n)$ with zero mean and variance $\sigma_x^2 = 1.0$ are used :

- White Gaussian noise
- White uniform noise
- Coloured Gaussian noise
- Logistic chaotic noise [99]
- Lorenz chaotic noise [9]
- Duffing chaotic noise [8]
- Stochastic nonlinear noise (named 'Nonlinear' in the tables)

The coloured Gaussian noise is the white Gaussian noise filtered by a 2nd order lowpass IIR filter with a normalised cut-off frequency of 0.05. The coloured Gaussian noise is normalised to have zero mean and unit

variance.

The stochastic nonlinear dynamic process is given as

$$\begin{aligned} x(n) = & [0.8 - 0.5 \exp(-(x(n-1))^2)]x(n-1) - \\ & [0.3 + 0.9 \exp(-(x(n-1))^2)]x(n-2) + \\ & 0.1 \sin(\pi x(n-1)) + v(n) \end{aligned} \quad (6.27)$$

Where $v(n)$ is white Gaussian process with zero mean and a variance of 0.01. The stochastic nonlinear noise is normalised to have zero mean and unit variance. The two dimensional attractor is shown in Figure 6.19.

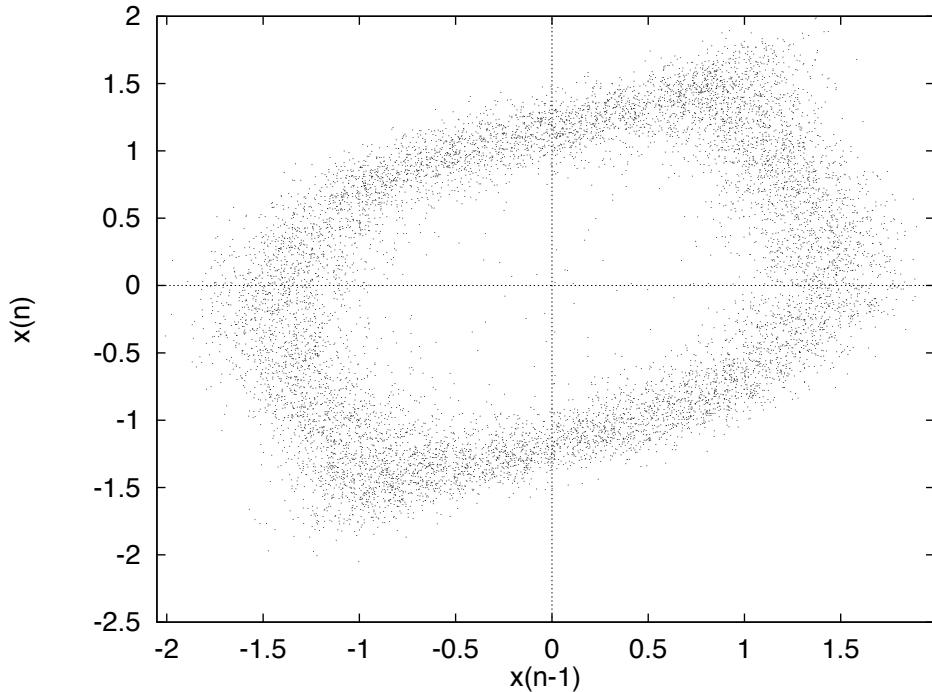


Figure 6.19: Attractor of the two dimensional stochastic nonlinear dynamical time series

For comparison in performance five different controllers were investigated.

- Linear FIR filter with $N = 10$ coefficients
- Volterra (quadratic, cubic) filter
- NRBF network ($L_{Logistic} = 337, L_{Lorenz} = 159, L_{Duffing} = 126, L_{Nonlinear} = 71$)
- Combined linear ($N = 10$) and Volterra (quadratic, cubic) filter
- Combined linear ($N = 10$) and NRBF network ($L_{Logistic} = 337, L_{Lorenz} = 159, L_{Duffing} = 126, L_{Nonlinear} = 71$)

The Tables 6.1 - 6.10 present the simulation results with different input signals $x(n)$ and actuators $H_a(z)$. The input vector $\mathbf{x}_N[n]$ has a dimension of $N = 10$ in Table 6.1. In Tables 6.2 - 6.5 the dimension M is 3 and in

Tables 6.6 -6.10 M is 5. The main acoustic plant $P(z)$ is a fourth-order FIR filter [177]:

$$P(z) = z^{-2} - 0.3z^{-3} + 0.2z^{-4}$$

The controller $W(z)$ does not depend on the error path transfer function $H_e(z)$. Therefore, only one error path $H_e(z)$ is used:

$$H_e(z) = z^{-5} \quad (6.28)$$

Three actuators $H_a(z)$ with different phase characteristics were chosen to test the theory in Section 6.3.3. The maximum phase transfer function of $H_{a1}(z)$:

$$H_{a1}(z) = 0.5 + z^{-1} \quad (6.29)$$

The minimum phase transfer function of $H_{a2}(z)$:

$$H_{a2}(z) = 1.0 + 0.5z^{-1} \quad (6.30)$$

The nonminimum phase transfer function of $H_{a3}(z)$:

$$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2} \quad (6.31)$$

The inverses of all combinations of $H_a(z)H_e(z)$ are estimated by exciting white noise through $H_a(z)H_e(z)$ and training an adaptive 32nd order FIR filter. The inverse modelling delay z^{-m} is $m = 16$ for $H_{a1}(z)H_e(z)$ and $H_{a3}(z)H_e(z)$ and $m = 5$ for $H_{a2}(z)H_e(z)$.

The least squares algorithm is a block least squares algorithm using the Householder transformation. The normalised mean square error (NMSE) is computed, after convergence, as follows :

$$\text{NMSE} = 10 \log_{10} \left(\frac{\sigma_e^2}{\sigma_d^2} \right) \quad \text{dB}$$

where σ_e^2 is the variance [197] of $e(n)$ and σ_d^2 is the variance of $d(n)$.

In the simulation where the input signal $x(n)$ is white Gaussian, white uniform or coloured Gaussian a purely linear controller achieves the best result. The results are provided in Table 6.1. The results for the white noise processes are provided as a basis for comparison since no linear or nonlinear predictive component is possible. When coloured Gaussian noise is used linear prediction improves the performance for the maximum phase actuator $H_{a1}(z)$ and nonminimum phase actuator $H_{a3}(z)$. The results for $H_{a2}(z)$ with a coloured input demonstrates a marked improvement in performance. At first sight, this may appear surprising. The actuator is minimum phase and thus no predictive component is required. However this is not a straightforward system identification task. The achievable NMSE is dependent on both the spectrum of the input signal and the transfer function of the actuator. Therefore there is no reason to expect the elements of column 4 in Table 6.1 to be the same as in the columns 2 and 3.

In Table 6.2 the weights in the nonlinear part of the Volterra filter converge to zero and therefore the combined controller approximates a linear model. The Volterra filter used here with only the quadratic and cubic terms

is inherently nonlinear and is not suited to produce a linear model. It has, therefore, the worst performance, as shown in Table 6.4. The NRBF network benefits strongly of having an additional linear part in Table 6.3, compared to the results in Table 6.5. From these results it can be seen that, if the noise $x(n)$ is an iid process, no matter what kind of distribution it has, or if the noise $x(n)$ is a coloured linear process, there are no benefits in using a nonlinear controller at all.

The situation changes dramatically when the input signal $x(n)$ is non-Gaussian and deterministic. Both combined linear and nonlinear controllers in Tables 6.9 and 6.10 achieve far better performances in conjunction with the maximum phase actuator $H_{a1}(z)$ and the nonminimum phase actuator $H_{a3}(z)$ compared to the linear controller in Table 6.6. The results for the input signal generated by the Duffing equation are initially surprising especially with respect to the minimum phase actuator $H_{a2}(z)$. However the architecture of (6.25) only provides a FIR approximation to $H_{a2}^{-1}(z)$. The additional terms could be provided through backward prediction using the embedding vector $\mathbf{x}_N(n)$. The architecture is flexible enough to provide this prediction implicitly without interference from the user.

Actuator $H_a(z)$	Linear		
	Gauss	Uniform	Colour
$H_{a1}(z) = 0.5 + z^{-1}$	-11	-11	-43
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-41	-41	-74
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-12	-12	-45

Table 6.1: NMSE in dB using a linear controller and stochastic noise for $x(n)$

Actuator H_a	Linear + Volterra		
	Gauss	Uniform	Colour
$H_{a1}(z) = 0.5 + z^{-1}$	-10.9	-11.0	-43
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-41	-41	-74
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-11.9	-12	-45

Table 6.2: NMSE in dB using a combined linear and nonlinear controller and stochastic noise for $x(n)$

Actuator H_a	Linear + NRBF		
	Gauss	Uniform	Colour
$H_{a1}(z) = 0.5 + z^{-1}$	-10.5	-10.9	-43
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-40.8	-40.9	-74
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-11.6	-11.9	-45

Table 6.3: NMSE in dB using a combined linear and nonlinear controller and stochastic noise for $x(n)$

Actuator H_a	Volterra		
	Gauss	Uniform	Colour
$H_{a1}(z) = 0.5 + z^{-1}$	-3.7	-6.2	-4.5
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-1.5	-2.0	-4.7
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-4.5	-6.5	-4

 Table 6.4: NMSE in dB using a nonlinear controller and stochastic noise for $x(n)$

Actuator H_a	NRBF		
	Gauss	Uniform	Colour
$H_{a1}(z) = 0.5 + z^{-1}$	-8.7	-8.8	-32
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-1.9	-2.0	-33
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-8.7	-8.7	-17.2

 Table 6.5: NMSE in dB using a nonlinear controller and stochastic noise for $x(n)$

Actuator H_{a(z)}	Linear			
	Logistic	Lorenz	Duffing	Nonlinear
$H_{a1}(z) = 0.5 + z^{-1}$	-10.8	-34.6	-70	-25.3
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-41	-64.8	-99.7	-55.3
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-12	-35.6	-66.6	-26.3

 Table 6.6: NMSE in dB using a linear controller and nonlinear noise for $x(n)$

Actuator H_a	Volterra			
	Logistic	Lorenz	Duffing	Nonlinear
$H_{a1}(z) = 0.5 + z^{-1}$	-16.8	-26.2	-23.5	-14.2
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-10.8	-27.9	-23.7	-13.9
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-17	-21.8	-20.3	-13.8

 Table 6.7: NMSE in dB using a nonlinear controller and nonlinear noise for $x(n)$

Actuator H_a	NRBF			
	Logistic	Lorenz	Duffing	Nonlinear
$H_{a1}(z) = 0.5 + z^{-1}$	-28.4	-55	-86.2	-28.2
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-10.3	-58	-91	-27.4
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-22.7	-43.3	-59.8	-28.5

 Table 6.8: NMSE in dB using a nonlinear controller and nonlinear noise for $x(n)$

Actuator $H_a(z)$	Linear + Volterra			
	Logistic	Lorenz	Duffing	Nonlinear
$H_{a1}(z) = 0.5 + z^{-1}$	-16.8	-50.5	-91	-28.7
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-41	-65.1	-105	-55.3
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-17.9	-50.5	-72.8	-29.7

 Table 6.9: NMSE in dB using a combined linear and nonlinear controller and nonlinear noise for $x(n)$

Actuator $H_a(z)$	Linear + NRBF			
	Logistic	Lorenz	Duffing	Nonlinear
$H_{a1}(z) = 0.5 + z^{-1}$	-28.5	-55.8	-89	-28.9
$H_{a2}(z) = 1.0 + 0.5z^{-1}$	-40.5	-65.6	-105.2	-55.3
$H_{a3}(z) = 1.0 + 1.5z^{-1} - z^{-2}$	-29.6	-53.5	-75	-29.7

 Table 6.10: NMSE in dB using a combined linear and nonlinear controller and nonlinear noise for $x(n)$

6.3.5 Discussion

This section is a more detailed analysis of the results from the input signal $x(n)$ generated by the Duffing equation. The controller W has to model $P(z)H_a(z)^{-1}$. Assuming that the controller has no difficulties in modelling $P(z)$ the error signal can be derived from the following block diagram in Figure 6.20.

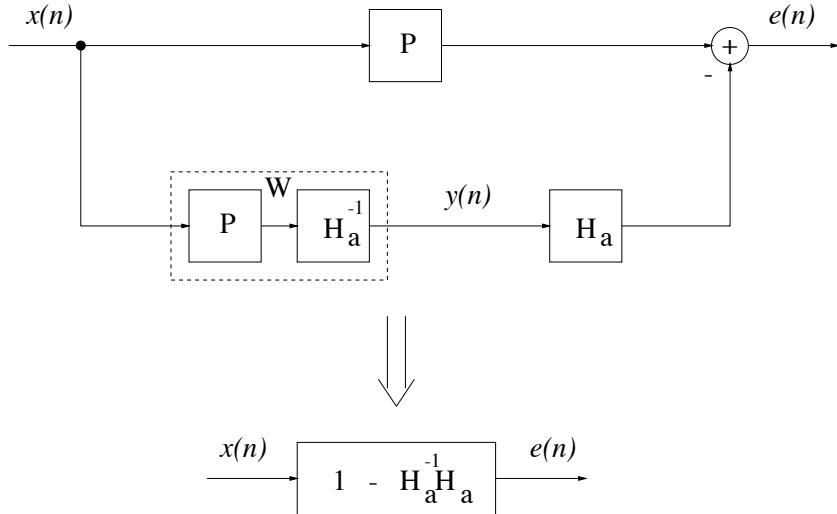


Figure 6.20: The influence of the input signal $x(n)$, controller $H_a(z)^{-1}$ and the nature of the actuator $H_a(z)$ on the error signal $e(n)$

The PSD of the error signal can be expressed as

$$S_{ee}(z) = \{1 - H_a(z)^{-1} H_a(z)\} \{1 - H_a(z^{-1})^{-1} H_a(z^{-1})\} S_{xx}(z) \quad (6.32)$$

Hence the error achieved will be dependent on the spectral characteristics of $x(n)$, the nature of $H_a(z)$ and its approximate inverse.

The simulations for the Duffing equation and the Logistic map as noise source were examined more closely. The NMSE in dB for the Duffing noise can be seen in Figure 6.21.

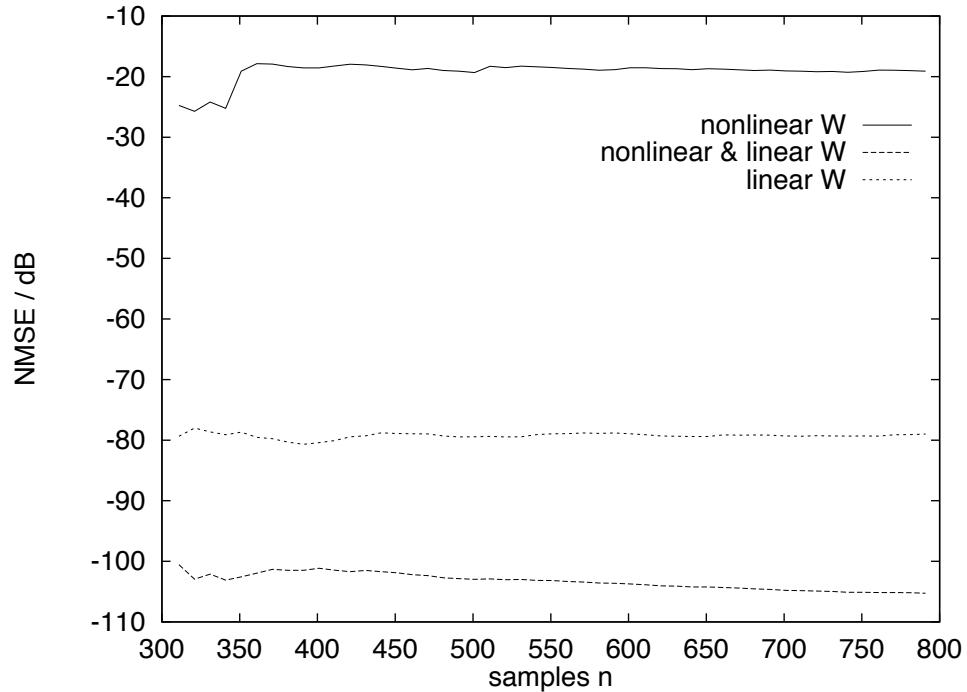


Figure 6.21: Duffing noise for $x(n)$, $M = 5$, $N = 5$, $H_a(z) = 1 + 0.5z^{-1}$, $H_e(z) = z^{-5}$, $m = 5$, 32 taps for $(\hat{H}_a \hat{H}_e)^{-1}$

Using a low order FIR controller shows the major improvement of the combined controller over the FIR controller in the NMSE. When using the Logistic noise source for $x(n)$ in Figure 6.22 the FIR controller produces, as expected, the best result in this linear control problem.

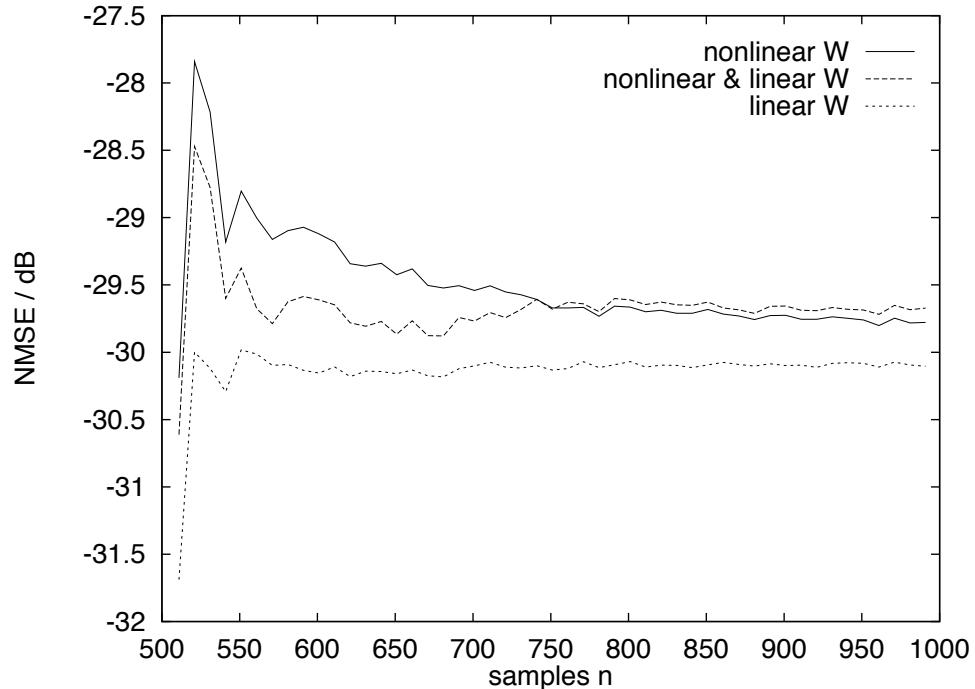


Figure 6.22: Logistic noise for $x(n)$, $M = 5$, $N = 5$, $H_a(z) = 1 + 0.5z^{-1}$, $H_e z = z^{-5}$, $m = 5$, 32 taps for $\left(\hat{H}_a \hat{H}_e\right)^{-1}$

The results from the Duffing equation become clear when the predictability of the chaotic time series is investigated. Figure 6.23 demonstrates the m-step predictability of the Duffing, Lorenz and Logistic time series used in the simulations. The predictability of the Duffing time series is extremely good and, therefore, explains the results obtained from previous simulations. A high order FIR filter will also produce good results, as it is able to approximate an IIR filter better. However, the delay caused by the high-order FIR filter may pose causality problems.

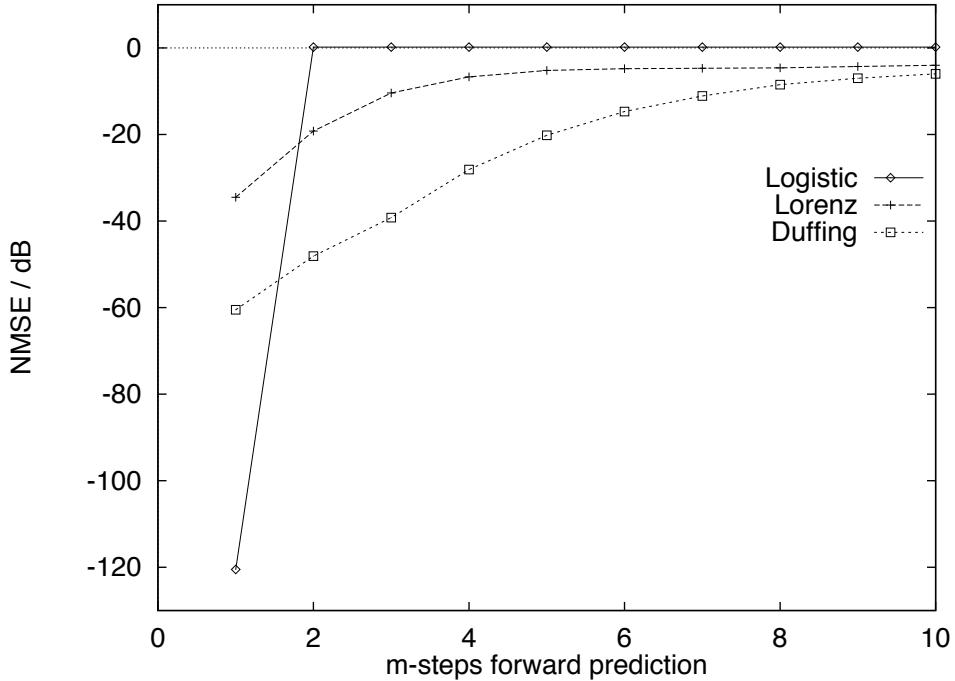


Figure 6.23: The m-step predictability of chaotic time series without feedback

6.4 Chapter Summary

This chapter gave an introduction to ANC. Two different control schemes, the feedback and the feedforward control approach were presented. The current state of the art of ANC systems in ducts was outlined. Different control schemes, such as the constrained and the modified filtered-x algorithms with the LMS and RLS algorithms were simulated.

A combined linear and nonlinear model is proposed for use in a feedforward ANC system in a duct. It has been shown that it is not possible to use the filtered-x LMS algorithm with a nonlinear controller, and therefore, the ANC scheme was redesigned. A mathematical expression has been derived for the controller and simulations have verified the theory. If the input signal $x(n)$ is nonlinear and deterministic, the nonlinear filters estimate $P(z)H_a^{-1}(z)$ more accurately than the traditional linear filter. This especially the case when $H_a(z)$ is nonminimum or maximum phase. A variety of simulation results with different models for $H_a(z)$ and $H_e(z)$ and noise signals have been presented and confirm the mathematical derivations of the combined linear and nonlinear controller.

Chapter 7

Conclusion

7.1 Review

The work presented in this thesis is primarily concerned with nonlinear noise cancellation. The noise or interference corrupting a signal of interest or a system is treated as a nonlinear and deterministic process rather than a stochastic process. Therefore, nonlinear filtering and control techniques are implemented in order to exploit the nonlinear structure in the observed time series. Obviously, nonlinear models are better suited for modelling or predicting nonlinear deterministic time series than linear models.

The nonlinear modelling capability of nonlinear models, such as RBF networks and VS filters, is exploited in a variety of different filter schemes. These filter schemes recover a narrowband signal which is corrupted by broadband noise. Conventional linear filtering techniques are not able to reduce the noise in the bandwidth of interest. Therefore, an improvement in performance can be achieved by applying nonlinear models to estimate the nonlinear deterministic noise in the bandwidth of interest. If the estimated nonlinear deterministic noise is subtracted from the corrupted signal of interest the SNR will improve.

The capability of predicting nonlinear deterministic time series is exploited in ANC. In this application the controller is confronted with the task of modelling a plant as well as the inverse transfer function of a loudspeaker and its driving device. The loudspeaker may be nonminimum phase which means that the inverse is a noncausal transfer function. Conventional linear controllers perform only sub-optimally in this scenario. However, nonlinear models are able to predict the nonlinear deterministic noise and, therefore, overcome the causality problem in an elegant way.

In Chapter 1 an introduction is given into the work presented in this thesis. The motivation for the research topic *Nonlinear Noise Cancellation* is laid out. The contributions of this thesis are given and an overview of the thesis ends the first chapter.

The basic concepts of nonlinear dynamics and chaos are presented in Chapter 2. Invariant measures which

classify a nonlinear dynamical system are discussed. Analysis tools for determining an appropriate sampling rate and embedding dimension are introduced and applied to chaotic time series. A brief review over Taken's delay embedding theorem is presented at the end of Chapter 2.

Nonlinear models are presented in Chapter 3. The chapter focuses on the VS filter and RBF network. These kind of nonlinear filters are *linear in their parameters* and, therefore, are easier to train than MLP networks. The topology and its training methods of the VS filter are presented. The ordinary or normalised RBF network is usually trained in two stages. The first stage is an unsupervised learning algorithm for positioning the kernels and estimating their bandwidths. The second stage is a supervised learning method which uses a desired signal to train the linear weights in the least square sense. Different unsupervised learning methods are discussed and their applicability to estimation problems in higher dimensions. An adaptive block least squares algorithm using the Householder transformation (see Appendix A) is used for training the linear weights.

An investigation into Broomhead's filter method is carried out in Chapter 4. The filter method recovers a narrowband signal from broadband noise. The filter method works well, as long as the FIR filter cancels the signal of interest completely. If it does not, the signal of interest will be modelled by the nonlinear model and, hence, decrease the performance. Another interesting fact is that linear filtering techniques when applied to the same experiments and the same restrictive assumptions perform almost as good or even better as Broomhead's filter method. It is further shown that high-order FIR filter which are able to attenuate the signal of interest considerably distort the structure of the low dimensional noise so much that the nonlinear model is not able to reconstruct the nonlinear deterministic noise.

To overcome the restrictions in Broomhead's filter method and to investigate into a more general approach a variety of filter methods are devised in Chapter 5. The so-called new filter method combines standard signal processing techniques to form a less restrictive filter scheme for cancelling broadband noise. The tricky part is to derive a reference and desired signal from the corrupted time series which are not correlated with each other from the signal of interest's point of view. Filterbanks capture the most important signal components for the two signals. The sampling rate is then dropped to re-embed the nonlinear deterministic time series and it also makes the scheme more computational efficient. An adaptive linear filter cancels the remaining linear correlation between the two signals and the nonlinear model is then trained to estimate the nonlinear deterministic noise. Simulation results over a range of input SNR with different filter techniques are presented. The new filter method improves the performance of the SNR compared to linear filtering techniques and overcomes the restriction of Broomhead's filter method.

Another filter method exploits the fact that a high-order FIR filter can be modelled by cascaded low-order FIR filters. The advantage is that low-order FIR filters do not disturb the structure of the nonlinear deterministic process too much. It is easier for the nonlinear models to undo the distortions caused by the low-order FIR filters. However, the nonlinear models produces only an estimate of the true nonlinear deterministic noise. Therefore, the performance drops from one cascade (FIR filter and nonlinear model) to the other and only performs sub-optimally overall when compared to linear filtering techniques.

The next filter method overcomes the problem of deriving a reference signal without the signal of interest present in a different way. It uses a quantiser which produces just enough quantisation noise to mask the signal

of interest in the reference signal. Simulations show that the nonlinear model is not able to detect the signal of interest and, therefore, only estimates the nonlinear deterministic noise. However, the additional quantisation noise blurs the structure of the nonlinear deterministic noise and the nonlinear models cause a greater estimation error. This greater estimation error effects the output SNR and in comparison with linear techniques this filter method only performs sub-optimally.

The problem of having the signal of interest present in the reference signal is reduced when the amplitude of the signal of interest is significantly smaller than the nonlinear deterministic noise. In this case only two FIR filters are necessary to provide the desired and reference signal for the nonlinear model. Compared with linear techniques this filter method is superior.

ANC in Chapter 6 is the other application, where the assumption is taken that the noise generated by a fan or engine can be modelled by a nonlinear deterministic process. An introduction into the principles of ANC and its physical limitations are presented. A background section outlines the current state-of-the-art of ANC in ducts. The most popular ANC scheme for feedforward control is the *filtered-x* LMS algorithm. Its derivation is presented and the need for an alternative control scheme for the implementation of a nonlinear controller is shown. Existing alternative schemes are simulated and compared. A control scheme which allows an adaptive block least squares approach is designed. A particular problem in ANC is the inversion process of the loudspeaker in the controller. The transfer function of the loudspeaker may be nonminimum phase and, therefore, has a noncausal inverse. This problem is overcome by using a combined linear and nonlinear model which is able to perform a linear system identification task as well as a nonlinear prediction task. The mathematical structure of the combined linear and nonlinear controller is derived and verified in simulations. A variety of noise sources in conjunction with different controllers are simulated. The combined linear and nonlinear controller shows a good improvement in performance compared to conventional controllers, when the noise is nonlinear and deterministic and the loudspeaker is nonminimum or maximum phase.

7.2 Observations

The filter schemes described in this thesis have actually a tough task. It is comparable with the blind equalisation problem in communication channels. There is no training signal available and the equaliser has to learn from the received data alone. The filter methods face the same dilemma, the signal of interest is unknown except for its location in the frequency domain. The challenge is to derive a reference signal from the corrupted signal which does not contain the signal of interest. In Broomhead's filter method this is given by the design of the FIR filter. The FIR filter is orthogonal to the signal of interest and completely removes it from the corrupted signal, signal of interest plus noise. However, if the signal of interest is more complex it is impossible to design such an orthogonal FIR filter.

There will be always a trade-off in the design of the FIR filters. On the one hand is desireable to have very short FIR filters which do not distort the structure of the nonlinear dynamic process too much. On the other hand the FIR filters have to be of a high-order to provide useful training signals for the nonlinear model.

7.3 Future Work

The obvious next task would be to run the simulations with real data. Real data always contains some stochastic noise and will cause problems for the nonlinear models in their estimation process. Hybrid models (linear/nonlinear) might be the answer to that problem. The linear part responsible of removing the stochastic noise and the nonlinear part for modelling or predicting the nonlinear deterministic noise.

Although, there is already evidence in the literature that noise processes can be low dimensional, nonlinear deterministic processes, it would be rather interesting to establish how valid this is over a range of different noise sources.

Another aspect to look into is the creation of parsimonious nonlinear models for practical implementation. The nonlinear models, because of their complexity need a lot of training data and training time compared to linear filtering or control techniques. Although DSP are becoming more faster and powerful it will be always an issue of finding parsimonious nonlinear models with fast training times.

The last filter scheme in Chapter 4 for recovering and detecting weak signals of interest in heavy nonlinear deterministic noise has not been fully investigated and exploited. For instance, it will be interesting to determine a boundary maximum input SNR for this method to work effectively. Also, the investigation to determine an optimum length for the FIR filters could gain further improvement in performance.

In the ANC scheme a nonlinear model could be incorporated into the loudspeaker model. It is expected that the developed combined linear and nonlinear controller will have better performance than the traditional linear control techniques. The hardware implementation of the combined controller in a DSP in conjunction with a experimental setup in a laboratory would be the ultimate goal.

Appendix A

Householder Transformation

The Householder transformation [127, 193] is a rotation and reflection technique which can be used to solve the least squares problem. The algorithm in this thesis is used in an adaptive block least squares approach. If an additional block of new data is available the least squares solution will be updated in a computational efficient way. Another advantage is that the algorithm is numerically robust, because it avoids estimating the inverse of the autocorrelation matrix directly.

The cost function over $k + 1$ data points to be minimised in the least square sense is defined as follows.

$$\xi(k) = \sum_{n=0}^k e^2(n) \quad (\text{A.1})$$

The error signal $e(n)$ is defined as

$$\begin{aligned} e(n) &= y(n) - \sum_{i=0}^{N-1} h_i x(n-i) \\ &= y(n) - \hat{y}(n) \end{aligned} \quad (\text{A.2})$$

where $\{x(n)\}$ is the input sequence, $\{h_n\}$ the weights of a FIR filter, N the number of weights and $\{y(n)\}$ is the desired signal.

Equation (A.2) can be rewritten using matrix and vector notations.

$$\mathbf{e}(k) = \mathbf{y}(k) - \mathbf{X}(k)\mathbf{h}(k) \quad (\text{A.3})$$

Where $\mathbf{X}(k)$ is the data matrix [127] of the input sequence.

To calculate the weights $\{h_n\}$ an autocorrelation matrix of the input data $\mathbf{R}_{xx}(k)$ and a crosscorrelation vector $\mathbf{r}_{xy}(k)$ are formed [127].

$$\mathbf{R}_{xx}(k)\mathbf{h}(k) = \mathbf{r}_{xy}(k) \quad (\text{A.4})$$

To avoid the inversion of the autocorrelation matrix following approach may be used. A matrix \mathbf{Q} which is orthonormal has two important properties. The first one is that the inverse can be found by a simple transpose operation

$$\mathbf{Q}^{-1} = \mathbf{Q}^T \quad (\text{A.5})$$

and the second one is

$$|\mathbf{e}| = |\mathbf{Q}\mathbf{e}| \quad (\text{A.6})$$

The matrix \mathbf{Q} reflects or rotates the vector \mathbf{e} without changing the Euclidean length of the vector \mathbf{e} . Therefore, the cost function can be written as

$$\xi(k) = |\mathbf{Q}(k)\mathbf{e}(k)| \quad (\text{A.7})$$

Using the matrix notation for describing the error vector \mathbf{e} yields to

$$\mathbf{Q}(k)\mathbf{e}(k) = \mathbf{Q}(k)\mathbf{y}(k) - \mathbf{Q}(k)\mathbf{X}(k)\mathbf{h}(k) \quad (\text{A.8})$$

The orthonormal transform of the data matrix \mathbf{X} yields to an upper triangular matrix

$$\mathbf{Q}(k)\mathbf{X}(k) = \begin{bmatrix} \mathbf{U}(k) \\ \mathbf{0} \end{bmatrix} \quad (\text{A.9})$$

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 to

$$\mathbf{Q}(k)\mathbf{y}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{v}(k) \end{bmatrix} \quad (\text{A.10})$$

The error vector can be now expressed as follows

$$\mathbf{Q}(k)\mathbf{e}(k) = \begin{bmatrix} \mathbf{u}(k) - \mathbf{U}(k)\mathbf{h}(k) \\ \mathbf{v}(k) \end{bmatrix} \quad (\text{A.11})$$

The Euclidean length of the error vector will be minimised when

$$\mathbf{u}(k) = \mathbf{U}(k)\mathbf{h}(k) \quad (\text{A.12})$$

which determines the weight vector $\mathbf{h}(k)$. The weights can be calculated by a simple back substitution operation, because $\mathbf{U}(k)$ is an upper triangular matrix. The minimum error is directly available by

$$|\mathbf{e}(k)| = |\mathbf{v}(k)| \quad (\text{A.13})$$

The unitary transformation of a data matrix into an upper triangular matrix may be achieved by using Householder transformations. Assume that \mathbf{R} is a rectangular data matrix whose first columns are already in upper triangular form. Thus \mathbf{R} may be partitioned as follows

$$\mathbf{R} = \begin{bmatrix} \mathbf{U} & * \\ \mathbf{0} & \mathbf{R}_S \end{bmatrix} \quad (\text{A.14})$$

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. Following matrix may be used

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_S \end{bmatrix} \begin{bmatrix} \mathbf{U} & * \\ \mathbf{0} & \mathbf{R}_S \end{bmatrix} = \begin{bmatrix} \mathbf{U} & * \\ \mathbf{0} & \mathbf{Q}_S \mathbf{R}_S \end{bmatrix} \quad (\text{A.15})$$

The unitary matrix \mathbf{Q}_S converts the first column of \mathbf{R}_S to zero elements with the exception of the first entry. The Householder transformation to determine the unitary matrix \mathbf{Q}_S is summarised below :

- A vector \mathbf{u} is defined

$$\mathbf{u} = \mathbf{r}_1 + \text{sign}(\mathbf{R}_S(1,1))|\mathbf{r}_1|\mathbf{1}$$

where $\mathbf{1} = [1, 0, 0, \dots, 0]^T$ and $\mathbf{R}_S(1,1)$ is the first column and first row. The sign function is defined as $\text{sign}(x) = 1$ if $x \geq 0$ and $\text{sign}(x) = -1$ otherwise.

- Use \mathbf{u} to define a vector $\hat{\mathbf{u}}$ for the transformation

$$\hat{\mathbf{u}} = \frac{2\mathbf{u}}{|\mathbf{u}|^2}$$

- Apply transformation to \mathbf{r}_1

$$\begin{aligned} \mathbf{Q}_S \mathbf{r}_1 &= \mathbf{r}_1 - \mathbf{u} \\ &= -\text{sign}(\mathbf{R}_S(1,1))|\mathbf{r}_1|\mathbf{1} \end{aligned}$$

The first element of $\mathbf{Q}_S \mathbf{r}_1$ is $-\text{sign}(\mathbf{R}_S(1,1))\sigma$ where $\sigma = \sqrt{\sum_i (\mathbf{R}_S(i,1))^2}$. The remaining elements are zero by definition.

- Apply the transform to the remaining columns, \mathbf{r}_j , of \mathbf{R}_S

$$\mathbf{Q}_S \mathbf{r}_j = \mathbf{r}_j - (\hat{\mathbf{u}}^T \mathbf{r}_j) \mathbf{u}$$

This algorithm can be used for finding and applying $\mathbf{Q}(k)$ for an adaptive block least squares solution.

Appendix B

Original Publications

The work described in this thesis has been reported in the following publications:

- † P. Strauch and B. Mulgrew, "Nonlinear Noise Cancellation using a Radial Basis Function Network", in Proceedings EUSIPCO'96, Trieste, Italy, vol. III, pages 1945-1948, September, 1996.
- † P. Strauch and B. Mulgrew, "Nonlinear Active Noise Control in a Linear Duct", in Proceedings ICASSP'97, Munich, Germany, vol. I, pages 395-398, April, 1997.

Paper submitted:

- P. Strauch and B. Mulgrew, "Active Control of Nonlinear Noise Processes in a Linear Duct", IEEE Transactions on Signal Processing.

† Reprinted in this appendix

References

- [1] D. S. Broomhead, J. P. Huke, and M. A. S. Potts, “Cancelling deterministic noise by constructing nonlinear inverses to linear filters”, *Physica D*, vol. 89, pp. 439–458, 1996.
- [2] D. S. Broomhead and G. P. King, “Extracting qualitative dynamics from experimental data”, *Physica 20D*, pp. 217–236, 1986.
- [3] A. M. Fraser, “Chaos and detection”, *Physical Review E*, vol. 53, no. 5, pp. 4514–4523, 1996.
- [4] D. S. Broomhead, J. P. Huke, and M. R. Muldoon, “Linear Filters and Nonlinear Systems”, *J. R. Statist. Soc. B*, vol. 54, no. 2, pp. 373, 1992.
- [5] G. Chen and X. Dong, “From chaos to order - perspectives and methodologies in controlling chaotic nonlinear dynamical systems”, *International Journal of Bif. and Chaos*, vol. 3, pp. 1363–1409, 1993.
- [6] H. D. I. Abarbanel, R. Brown, and J. B. Kadtke, “Prediction and system identification in chaotic nonlinear systems: Time series with broadband spectra”, *Physics Letters A*, vol. 138, no. 8, pp. 401, July 1989.
- [7] A. M. Fraser, “Modeling nonlinear time series”, *ICASSP Proceedings IEEE*, vol. 5, pp. 313, 1992.
- [8] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*. Springer-Verlag, 1989.
- [9] J. P. Eckmann and D. Ruelle, “Ergodic theory of chaos and strange attractors”, *Reviews of Modern Physics*, vol. 57, no. 3/1, pp. 617, July 1985.
- [10] J. K. Kanters and N. H. Holstein-Rathlou, “Heart rate variability is not due to chaos”, *Journal of Ambulatory Monitoring*, 1994.
- [11] N. A. Gershenfeld and A. S. Weigend, *The Future of Time Series: Learning and Understanding, in Predicting the Future and Understanding the Past: a Comparison of Approaches*. Addison-Wesley, 1993.
- [12] M. Bernardo, F. Garofalo, L. Glielmo, and F. Vasca, “Impacts, bifurcations and chaos in the dc/dc buck converter”, *to appear in IEEE Transactions on Circuits and Systems*.
- [13] Y. Hayakawa, A. Marumoto, and Y. Sawada, “Effects of the chaotic noise on the performance of a neural network model for optimization problems”, *Physical Review E*, vol. 51, no. 4, pp. 2693–2696, April 1995.
- [14] T. W. Frison and H. D. I. Abarbanel et al., “Chaos in ocean ambient ”noise””, *J. Acoust. Soc. Am.*, vol. 99, no. 3, pp. 1527–1539, March 1996.
- [15] W. W. Taylor, *Application of nonlinear prediction to signal separation, in : Nonlinear Modelling and Forecasting*. Addison-Wesley, Reading MA, 1992.
- [16] L. O. Chua et al., “Special Issue on Chaos in Nonlinear Electronic Circuits”, *IEEE Transactions on Circuits and Systems I & II*, vol. 40, no. 10, 1993.
- [17] M. Banbrook. *Nonlinear analysis of speech from a synthesis perspective*. PhD thesis, The University of Edinburgh, 1996.
- [18] H. Leung and S. Haykin, “Is there a radar clutter attractor ?”, *Appl. Phys. Letters*, vol. 56, pp. 593–595, 1990.

- [19] T. Matsuura, T. Hiei, H. Itoh, and K. Torikoshi, “Active noise control by using prediction of time series data with a neural network”, *SMC IEEE Proceedings*, vol. 3, pp. 2070–2075, 1995.
- [20] S. Hayes, C. Grebogi, and E. Ott, “Communicating with chaos”, *Phys. Rev. Lett.*, vol. 70, pp. 3031–3034, 1993.
- [21] G. W. Flake, G. Z. Sun, Y. C. Lee, and H. H. Chen, “Exploiting chaos to control the future”, *NIPS'93, Denver*, 1993.
- [22] D. T. Kaplan and L. Glass, “Direct test for determinism in a time series”, *Physical Review Letters*, vol. 68, no. 4, pp. 427, January 1992.
- [23] A. E. Jaquin, “Image coding based on a fractal theory of iterated contractive image transformations”, *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 18–30, 1992.
- [24] Becker and Doerfler, *Dynamical systems and fractals*. Cambridge University Press, 1990.
- [25] H. Leung and X. Huang, “Parameter estimation in chaotic noise”, *IEEE Transactions on Signal Processing*, vol. 44, no. 10, pp. 2456–2463, 1996.
- [26] H. D. I. Abarbanel, R. Brown, and J. B. Kadkne, “Prediction in chaotic nonlinear systems: Methods for time series with broadband fourier spectra”, *Physical Review A*, vol. 41, no. 4, pp. 1782, February 1990.
- [27] J. D. Farmer and J. J. Sidorowich, “Predicting chaotic time series”, *Physical Review Letters*, vol. 59, pp. 845, August 1987.
- [28] A. C. Singer, G. W. Wornell, and A. V. Oppenheim, “Codebook prediction: A nonlinear signal modeling paradigm”, *ICASSP Proceedings IEEE*, vol. 5, pp. 325, 1992.
- [29] M. Casdagli, “Nonlinear prediction of chaotic time series”, *Physica D*, vol. 35, pp. 335, June 1988.
- [30] S. M. Cox and A. J. Roberts, “Initial conditions for models of dynamical systems”, *Physica D (preprint)*, 1994.
- [31] Q. Yao and H. Tong, “Quantifying the influence of initial values on nonlinear prediction”, *J. R. Statist. Soc. B*, vol. 56, no. 4, pp. 701–725, 1994.
- [32] G. Sugihara and R. M. May, “Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series”, *Nature*, vol. 344, pp. 734, April 1990.
- [33] G. L. Baker and J. P. Gollub, *Chaotic Dynamics*. Cambridge, 1990.
- [34] D. J. Bell, *Mathematics of Linear and Nonlinear Systems*. Oxford, 1991.
- [35] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer-Verlag, 1990.
- [36] T. S. Parker and L. O. Chua, “Chaos: A tutorial for engineers”, *Proceedings of the IEEE*, vol. 75, no. 8, pp. 979, August 1987.
- [37] J. G. Proakis and C. M. Rader, *Advanced Digital Signal Processing*. Maxwell Macmillan International Editions, 1992.
- [38] P. Flandrin and O. Michel, “Higher-order within chaos”, *IEEE Signal Processing Workshop on Higher-Order Statistics*, pp. 295, June 1993.
- [39] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectra Analysis*. Prentice Hall, 1993.
- [40] T. S. Rao, *Analysis of nonlinear time series (and chaos) by bispectral methods*. Nonlinear Modeling and Forecasting. Addison-Wesley, 1992.
- [41] G. R. Wilson and W. Ellinger, “Detection of chaotic processes in Gaussian noise at low signal to noise ratios using higher order spectra”, *IEEE Signal Processing Workshop on Higher Order Statistics*, 1993.

- [42] C. Pezeshki, S. Elgar, and R. C. Krishna, “Bispectral analysis of possessing chaotic motion”, *Journal of Sound and Vibration*, vol. 137, pp. 357–368, 1990.
- [43] J. W. A. Fackrell, P. R. White, J. K. Hammond, and R. J. Pinnington, “The interpretation of the bispectra of vibration signals-I. theory”, *Mechanical Systems and Signal Processing*, vol. 9, pp. 257–266, 1995.
- [44] J. W. A. Fackrell, P. R. White, J. K. Hammond, and R. J. Pinnington, “The interpretation of the bispectra of vibration signals-II. experimental results and applications”, *Mechanical Systems and Signal Processing*, vol. 9, pp. 267–274, 1995.
- [45] C. Pezeshki, S. Elgar, R. Krishna, and T. D. Burton, “Auto and cross-bispectral analysis of a system of two coupled oscillators with quadratic nonlinearities possessing chaotic motion”, *Journal of Applied Mechanics*, vol. 59, pp. 657–662, 1992.
- [46] C. Cutler and D. Kaplan, *Nonlinear dynamics and time series : building a bridge between the natural and statistical sciences*. Providence, R.I. : American Mathematical Society, 1997.
- [47] R. C. Hilborn, *Chaos and nonlinear dynamics : an introduction for scientists and engineers*. Oxford University Press, 1994.
- [48] S. H. Strogatz, *Nonlinear dynamics and chaos : with applications to physics, biology, chemistry, and engineering*. Addison-Wesley, 1994.
- [49] H. O. Peitgen, H. Juergens, and D. Saupe, *Chaos and Fractals*. Springer Verlag, 1992.
- [50] M. Casdagli, “Chaos and deterministic versus stochastic nonlinear modelling”, *J. R. Statist. Soc. B*, vol. 54, no. 2, pp. 303, 1992.
- [51] H. D. I. Abarbanel, R. Brown, J. J. Sidowich, and L. S. Tsimring, “The analysis of observed chaotic data in physical systems”, *Reviews of Modern Physics*, vol. 65, no. 4, pp. 1331, October 1993.
- [52] R. L. Smith, “Estimating dimension in noisy chaotic time series”, *J. R. Statist. Soc. B*, vol. 54, no. 2, pp. 329, 1992.
- [53] P. Grassberger and I. Procaccia, “Characterization of strange attractors”, *Physical Review Letters*, vol. 50, no. 5, pp. 346–349, 1983.
- [54] A. M. Albano, J. Munech, and C. Schwartz, “Singular-value decomposition and the Grassberger-Procaccia algorithm”, *Physical Review A*, vol. 38, no. 6, pp. 3017, September 1988.
- [55] M. B. Kennel, H. D. I. Abarbanel, and J. J. Sidorowich. Prediction errors and local lyapunov exponents. Mail Code 0402, La Jolla, CA 92093-0402, March 1994.
- [56] R. C. L. Wolff, “Local lyapunov exponents: Looking closely at chaos”, *J. R. Statist. Soc. B*, vol. 54, no. 2, pp. 353, 1992.
- [57] H. D. I. Abarbanel, R. Brown, and M. B. Kennel, “Local lyapunov exponents computed from observed data”, *Journal of Nonlinear Science*, vol. 1, pp. 175–199, 1991.
- [58] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, “Geometry from a time series”, *Physical Review Letters*, vol. 45, no. 9, pp. 712, September 1980.
- [59] B. Cheng and H. Tong, “On consistent nonparametric order determination and chaos”, *J. R. Statist. Soc. B*, vol. 54, no. 2, pp. 427, 1992.
- [60] F. Takens, “Detecting strange attractors in turbulence”, *Lecture Notes in Mathematics*, no. 898, pp. 366, 1988. Dynamical Systems and Turbulence, Warwick 1980, Springer Verlag.
- [61] T. Sauer, J. A. Yorke, and M. Casdagli, “Embedology”, *Journal of Statistical Physics*, vol. 65, pp. 579, 1991.
- [62] J. F. Gibson, J. D. Farmer, M. Casdagli, and S. Eubank, “An analytic approach to practical state space reconstruction”, *Physica D*, no. 57, pp. 1–30, 1992.

- [63] R. Cawley and G. H. Hsu, "Local-geometric-projection method for noise reduction in chaotic maps and flows", *Physical Review A*, vol. 46, no. 6, pp. 3057, September 1992.
- [64] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction", *Physical Review A*, vol. 45, no. 6, pp. 3403, March 1992.
- [65] M. B. Kennel and H. D. I. Abarbanel, "False neighbors and false strands: A reliable minimum embedding dimension algorithm", ftp://lyapunov.ucsd.edu/pub/nonlinear/false_strands.tar.gz, October 1994.
- [66] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information", *Physical Review A*, vol. 33, no. 2, pp. 1134–1140, February 1986.
- [67] I. Scott and B. Mulgrew, "Nonlinear system identification and prediction using orthonormal functions", *IEEE Transaction on Signal Processing*, 1997. to appear in July.
- [68] J. Thyssen, H. Nielsen, and S. D. Hansen, "Nonlinear short-term prediction in speech coding", *ICASSP'94*, vol. 1, pp. 185–188, 1994.
- [69] J. Thyssen, H. Nielsen, and S. D. Hansen, "Quantization of nonlinear predictors in speech coding", *ICASSP'95*, pp. 265–268, 1995.
- [70] V. Tresp and R. Hofmann, "Missing and noisy data in nonlinear time series prediction", *Neural Networks for Signal Processing 5*, 1995.
- [71] M. Schetzen, "Nonlinear system modeling based on the Wiener theory", *Proceedings of the IEEE*, vol. 69, no. 12, pp. 1557, December 1981.
- [72] S. A. Billings, "Identification of nonlinear systems - a survey", *IEE Proceedings*, vol. 127, no. 6, pp. 272, November 1980.
- [73] R. W. Lucky, "Modulation and detection for data transmission on the telephone channel", *New directions in signal processing in communication and control*, 1975. J. K. Skwirzynski, ed., Leiden, Holland, Noordhoff.
- [74] J. Thyssen, H. Nielsen, and S. D. Hansen, "Nonlinearities in speech", *NSIP - 95*, vol. 2, pp. 662–665, 1995.
- [75] D. R. Hush and B. G. Horne, "Progress in supervised neural networks", *IEEE Signal Processing Magazine*, January 1993.
- [76] D. F. Specht, "Probabilistic neural networks", *Neural Networks*, vol. 3, pp. 109–118, 1990.
- [77] W. S. Sarle, "Neural networks and statistical models", *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, April 1994.
- [78] J. S. R. Jang and C. T. Sun, "Neuro-fuzzy modeling and control", *IEEE Proceedings*, March 1995.
- [79] C. T. Wangler and C. H. Hansen, "Genetic algorithm adaption of non-linear filter structures for active sound and vibration control", *ICASSP'94*, vol. 3, pp. 505–508, 1994.
- [80] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons - From backpropagation to adaptive learning algorithms", *Computer Standards and Interfaces, Special Issue on Neural Networks*, vol. 5, 1994.
- [81] T. Koh and E. J. Powers, "Second-order volterra filtering and its application to nonlinear system identification", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 6, pp. 1445, December 1985.
- [82] S. Boyd and L. O. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series", *IEEE Transactions on Circuits and Systems*, vol. CAS-32, no. 11, pp. 1150, November 1985.
- [83] F. M. A. Acosta, "Radial basis function and related models: An overview", *Signal Processing*, no. 45, pp. 37–58, 1995.

- [84] S. A. Billings and C. F. Fung, “Recurrent radial basis function networks for adaptive noise cancellation”, *Neural Networks*, vol. 8, no. 2, pp. 273–290, 1995.
- [85] L. Tarassenko and S. Roberts, “Supervised and unsupervised learning in radial basis function classifiers”, *IEE Proceedings - Vis. Image Signal Processing*, vol. 141, no. 4, August 1994.
- [86] I. Cha and S. A. Kassam, “RBFN restoration of nonlinearly degraded images”, *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 964–975, June 1996.
- [87] M. R. Berthold and J. Diamond, “Boosting the performance of RBF networks with dynamic decay adjustment”, *Advances in Neural Information Processing*, vol. 7, 1995.
- [88] S. Haykin, *Neural Networks*. MACMILLAN, 1994.
- [89] V. J. Mathews, “Adaptive polynomial filters”, *IEEE Signal Processing Magazine*, July 1991.
- [90] I. W. Sandberg, “The mathematical foundations of associated expansions for mildly nonlinear systems”, *IEEE Transactions on Circuits and Systems*, vol. Cas-30, pp. 441, July 1983.
- [91] G. B. Giannakis and A. V. Dandawate, “Linear and non-linear adaptive noise cancelers”, *ICASSP’90*, vol. 3, pp. 1373–1376, 1990.
- [92] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [93] M. J. D. Powell, “Radial basis functions for multivariable interpolation: a review”, *Algorithms for Approximation*, pp. 143–167, 1987. J. C. Mason and M. G. Cox, Eds., Oxford.
- [94] S. Chen, C. F. N. Cowan, and P. M. Grant, “Orthogonal least squares learning algorithm for radial basis function networks”, *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, March 1991.
- [95] J. A. Leonard and M. A. Kramer, “Radial basis function networks for classifying process faults”, *IEEE Control Systems*, April 1991.
- [96] Q. Zhao and Z. Bao, “Radar target recognition using a radial basis function neural network”, *Neural Networks*, vol. 9, no. 4, pp. 709–720, 1996.
- [97] A. Krzyzak, T. Linder, and G. Lugosi, “Nonparametric estimation and classification using radial basis function nets and empirical risk minimization”, *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 475–487, March 1996.
- [98] S. Elanayar and Y. C. Shin, “Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems”, *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 594, July 1994.
- [99] D. S. Broomhead and D. Lowe, “Multivariable functional interpolation and adaptive networks”, *Complex System*, vol. 2, pp. 321, 1988.
- [100] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge University Press, 1996.
- [101] I. Cha and S. A. Kassam, “Interference cancellation using radial basis function networks”, *Signal Processing*, vol. 47, pp. 247–268, 1995.
- [102] D. Obradovic, “On-line training of recurrent neural networks with continuous topology adaption”, *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 222–228, January 1996.
- [103] W. S. Sarle, “Stopped training and other remedies for overfitting”, *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, pp. 352–360, 1995.
- [104] C. Wang. *A theory of generalisation in learning machines with neural network application*. PhD thesis, University of Pennsylvania, 1994.
- [105] E. S. Chng. *Applications of nonlinear filters with the linear-in-the-parameter structure*. PhD thesis, University of Edinburgh, 1995.

- [106] A. Sherstinsky and R. W. Picard, “On the efficiency of orthogonal least squares training method for radial basis function networks”, *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 195–200, 1996.
- [107] Warren Sarle et al. Neural network FAQ. <ftp://ftp.sas.com/pub/neural/FAQ.html>.
- [108] P. V. Balakrishnan, M. C. Cooper, V. S. Jacob, and P. A. Lewis, “A study of the classification capabilities of neural networks using unsupervised learning: A comparison with k-means clustering”, *Psychometrika*, vol. 59, pp. 509–525, 1994.
- [109] J. Laaksonen, “Clasification with learning k-nearest neighbours”, *ICNN'96*, pp. 1480–1483, 1996.
- [110] S. Chen, “Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning”, *IEE Electronic Letters*, vol. 31, no. 2, pp. 117, January 1995.
- [111] C. Chinrungrueng and C. H. Sequin, “Optimal adaptive k-means algorithm with dynamic adjustment of learning rate”, *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 157, January 1995.
- [112] R. A. Redner H. F. Walker, “Mixture densities, maximum likelihood and the EM algorithm”, *SIAM Review*, vol. 26, no. 2, pp. 195–239, April 1984.
- [113] A. Ukrainec and S. Haykin, “Signal processing with radial basis function networks using expectation maximization algorithm clustering”, *SPIE Adaptive Signal Processing*, vol. 1565, pp. 529–539, 1991.
- [114] D. W. Scott, *Multivariate density estimation : theory, practice, and visualization*. Wiley, 1992.
- [115] R. Orre and A. Lansner, “Pulp quality modelling using Bayesian mixture density neural networks”, *Journal of Systems Engineering*, no. 6, pp. 128–136, 1996.
- [116] R. Shorten and R. Murray-Smith, “Side effects of normalising radial basis function networks”, *International Journal of Neural Systems*, vol. 7, no. 2, pp. 167–179, May 1996.
- [117] D. S. Broomhead. Nonlinear algorithms. IEATP Project 3119 DRA (Malvern), June 1993.
- [118] E. J. Kostelich and J. A. Yorke, “Noise reduction in dynamical systems”, *Physical Review A*, vol. 38, no. 3, pp. 1649, August 1988.
- [119] D. Nychka, S. Ellner, A. R. Gallant, and D. McCaffrey, “Finding chaos in noisy systems”, *J. R. Statist. Soc. B*, vol. 54, no. 2, pp. 399, 1992.
- [120] D. S. Broomhead, J. P. Huke, and R. Jones, “Signals in chaos: A method for the cancellation of deterministic noise from discrete signals”, *Physica D*, vol. 80, pp. 413–432, 1995.
- [121] J. Stark and B. V. Arumugam, “Extracting slowly varying signals from a chaotic background”, *Int. J. Bif. Chaos*, vol. 2, pp. 413–419, 1992.
- [122] R. Badii, G. Broggi, B. Derighetti, and M. Ravani, “Dimension increase in filtered chaotic signals”, *Physical Review Letters*, vol. 60, no. 11, pp. 979, March 1988.
- [123] F. Mitschke, M. Moeller, and W. Lange, “Measuring filtered chaotic signals”, *Physical Review A*, vol. 37, no. 11, pp. 4518, June 1988.
- [124] M. E. Davies and K. M. Cambell, “Linear recursive filters and nonlinear dynamics”, *Nonlinearity*, vol. 9, pp. 487–499, 1996.
- [125] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. Macmillan, 2 ed., 1992.
- [126] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge, 1990.
- [127] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2 ed., 1991.
- [128] S. M. Kuo and M. J. Ji, “Development and analysis of an adaptive noise equalizer”, *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 3, pp. 217–222, 1995.

- [129] S. M. Kuo and M. J. Ji, "Principle and application of adaptive noise equaliser", *IEEE Transactions on Circuits and Systems II*, vol. 41, no. 7, pp. 471–474, 1994.
- [130] A. V. Oppenheim and R. W. Schafer, *Digital signal processing*. Prentice-Hall, 1975.
- [131] R. G. Shenoy, D. Burnside, and T. W. Parks, "Linear periodic systems and multirate filter design", *IEEE Transactions on Signal Processing*, vol. 42, no. 9, pp. 2242–2255, 1994.
- [132] V. P. Sathe and P. P. Vaidyanathan, "Effects of Multirate Systems on the Statistical Properties of Random Signals", *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 131–146, January 1993.
- [133] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice-Hall, 1983.
- [134] N. Fliege, *Multirate digital signal processing : multirate systems, filterbanks, wavelets*. John Wiley, 1994.
- [135] N. S. Jayant and P. Noll, *Digital Coding Of Waveforms*. Prentice-Hall, 1984.
- [136] D. S. Broomhead, J. P. Huke, and R. Jones, "Signals in chaos: Methods for the cancellation of deterministic noise", *IEATP Project 3119 Nonlinear Algorithms, Working Note 7*, 1993.
- [137] S. M. Hammel, "A noise reduction method for chaotic systems", *Phys. Letts. A*, vol. 148, pp. 421–428, 1990.
- [138] P. A. Nelson and S. J. Elliott, *Active Control of Sound*. Academic Press INC., 3rd ed., 1995.
- [139] H. K. Pelton, S. Wise, and W. S. Sims, "Active HVAC noise control systems provide acoustical comfort", *Sound and Vibration*, pp. 14–18, July 1994.
- [140] B. Sanito, "Electronic automobile muffler quiets the skeptics", *EE Times*, October 1992.
- [141] A. J. Salloway and C. E. Millar, "Active vibration and noise control", *GEC Review*, vol. 11, no. 3, pp. 138–145, 1996.
- [142] J. Aplin, "Active noise control - from research to reality", *AIAA IEEE Digital Avionics System Conference 13th DASC*, 1994.
- [143] S. J. Elliott and P. A. Nelson, "Active Noise Control", *IEEE Signal Processing Magazine*, October 1993.
- [144] T. Auspitzer, D. Guicking, and S. J. Elliot, "Using a fast-recursive-least-squared algorithm in a feedback-controller", *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 61–64, 1995.
- [145] S. J. Elliot and T. J. Sutton, "Performance of feedforward and feedback systems for active control", *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 3, pp. 214–223, 1996.
- [146] C. E. Ruckmann and C. R. Fuller, "A regression approach for simulating feedforward active noise control", *J. Acoust. Soc. Am.*, vol. 97, no. 5, pp. 2906–2918, 1995.
- [147] B. Rafaely and M. Furst, "Audiometric ear canal probe with active ambient noise control", *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 3, pp. 224–230, May 1996.
- [148] R. Shoureshi, L. Brackney, N. Kubota, and G. Batta, "A modern control approach to active noise control", *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 673–678, December 1993.
- [149] E. Benzaria and V. Martin, "Secondary source locations in active noise control: Selection or optimization?", *Journal of Sound and Vibration*, vol. 173, pp. 137–144, 1994.
- [150] A. Gonzalez and S. J. Elliot, "Adaptive minimization of the maximum error signal in an active control system", *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 53–56, 1995.
- [151] M. O. Tokhi, M. A. Hossain, and K. Mamour, "Self-tuning active control of noise and vibration", *International Conference on Control*, vol. 1, pp. 771–776, 1994.

- [152] P. A. Nelson, F. Orduna-Bustamante, and H. Hamada, “Inverse filter design and equalization zones in multichannel sound reproduction”, *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 3, pp. 185–192, 1995.
- [153] O. Kipersztok and R. Hammond, “Fuzzy active control of a distributed broadband noise source”, *Proc. of the Third IEEE Conference on Fuzzy Systems*, vol. 2, pp. 1342–1347, 1994.
- [154] S. J. Elliot and P. A. Nelson. Models for describing active noise control in ducts. Technical Report 127, Institute of Sound and Vibration Research, April 1984.
- [155] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics*. Wiley, 3 ed., 1982.
- [156] Z. Wu, V. K. Varadan, and V. V. Varadan K. Y. Lee, “A state-space modeling of one-dimensional active noise control systems”, *Transactions of the AMSE*, vol. 117, pp. 220–225, 1995.
- [157] J. Hong et al., “Modeling, identification, and feedback control of noise in an acoustic duct”, *IEEE Transactions on Control Systems Technology*, vol. 4, no. 3, pp. 283–291, May 1996.
- [158] A. J. Hull, C. J. Radcliffe, and S. C. Southward, “Global active noise control of a one-dimensional acoustic duct using a feedback controller”, *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 488–494, 1993.
- [159] T. C. Yang, C. H. Tseng, and S. F. Ling, “Constrained optimization of active noise control systems in enclosures”, *J. Acoust. Soc. Am.*, vol. 95, no. 6, pp. 3390–3399, 1994.
- [160] J. S. Hu, “Active noise cancellation in ducts using internal model-based control algorithms”, *IEEE Transactions on Control Systems Technology*, vol. 4, no. 2, pp. 163–170, March 1996.
- [161] J. B. Fahline, “Active control of the sound radiated by a vibrating body using only a layer of simple sources”, *J. Acoust. Soc. Am.*, vol. 97, no. 4, pp. 2249–2254, 1995.
- [162] Y. Okamoto, H. Boden, and M. Abom, “Active noise control in ducts via side-branch resonators”, *J. Acoust. Soc. Am.*, vol. 96, no. 3, pp. 1533–1538, 1994.
- [163] D. R. Morgan and C. Sanford, “A control theory approach to the stability and transient analysis of the filtered-x LMS adaptive notch filter”, *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2341–2346, 1992.
- [164] D. R. Morgan and J. Thi, “A multi-tone pseudo-cascade filtered-x LMS adaptive notch filter”, *ICASSP’91*, vol. 3, pp. 2093–2096, 1991.
- [165] E. Bjarnason, “Analysis of the filtered-x LMS algorithm”, *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, November 1995.
- [166] C. C. Boucher, S. J. Elliot, and P. A. Nelson, “Effect of errors in the plant model on the performance of algorithms for adaptive feedforward control”, *IEE Proceedings-F*, vol. 138, no. 4, pp. 313–319, August 1991.
- [167] S. D. Snyder and C. H. Hansen, “The effect of transfer function estimation errors on the filtered-x lms algorithm”, *IEEE Transactions on Signal Processing*, vol. 42, no. 4, pp. 950–953, 1994.
- [168] E. Bjarnason, “Analysis of the filtered-x LMS algorithm”, *ICASSP’93*, vol. 3, pp. 511–514, 1993.
- [169] E. Bjarnason, “Algorithms for active noise cancellation without exact knowledge of the error-path filter”, *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 573–576, 1994.
- [170] D. R. Morgan, “An analysis of multiple correlation cancellation loops with a filter in the auxillary path”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-28, pp. 454–467, 1980.
- [171] B. Widrow, D. Shur, and S. Schaffer, “On adaptive inverse control”, *Proceedings of the 15th ASILOMAR Conference on Circuits, Systems and Computers*, pp. 185–195, 1981.

- [172] J. C. Burgess, “Active adaptive sound control in a duct: a computer simulation”, *Journal of the Acoustical Society of America*, vol. 70, pp. 715–726, 1980.
- [173] K. S. Lee, J. C. Lee, and D. H. Youn, “A new algorithm for active cancellation of fan noise”, *IEEE International Conference on Consumer Electronics*, pp. 426–427, 1995.
- [174] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, 1985.
- [175] S. J. Flockton, “Fast adaption algorithms in active noise control”, *2nd Conference on Recent Advances in Active Control of Sound and Vibration*, pp. 802–810, 1993.
- [176] K. A. Chen and Y. L. Ma, “An intermittent RLS algorithm and its application to adaptive active wideband noise control”, *Proc. of the IEEE International Symposium on Industrial Electronics*, vol. 1, pp. 442–445, 1992.
- [177] I. S. Kim, H. S. Na, K. J. Kim, and Y. Park, “Constraint filtered-x and filtered-u least-mean-square algorithms for the active control of noise in ducts”, *J. Acoust. Soc. Am.*, vol. 95, no. 6, pp. 3379–3389, June 1994.
- [178] R. Schirmacher, T. Auspitzer, and D. Guicking, “Anwendung des SFAEST-Algorithmus in der aktiven Laermbelempfung”, *Fortschritte der Akustik - DAGA94*, pp. 513–516, 1994. (in german).
- [179] E. Bjarnason, “Active noise cancellation using a modified form of the filtered-x LMS algorithm”, *Signal Processing VI: Theories and Applications*, pp. 1053–1056, 1992.
- [180] D. Guicking and R. Schirmacher, “A broadband active noise control system using a fast RLS algorithm”, *Third International Congress on Air- and Structure-Borne Sound and Vibration, Montreal*, pp. 1361–1368, 1994.
- [181] R. Schirmacher, “Algorithmen und Modellierungskonzepte zur aktiven Schallfeldbeeinflussung mit Feed-Forward-Systemen”, *Fortschritte der Akustik - DAGA96*, 1996.
- [182] R. Schirmacher and D. Guicking, “ARMA-Modelle in der aktiven Laermkompenstation und ihre Realisierung durch schnelle adaptive Filter”, *Fortschritte der Akustik DAGA-95*, pp. 503–506, 1995.
- [183] R. Schirmacher, “Experimentelle Untersuchung zu einem aktiven Absorber auf der Basis schneller adaptiver IIR-Filter”, *Fortschritte der Akustik - DAGA96*, 1996.
- [184] S. D. Snyder, “Active control using IIR filters - a second look”, *ICASSP'94*, vol. 2, pp. 241–244, 1994.
- [185] P. R. Chang and B. F. Yeh, “Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks”, *J. Acoust. Soc. Am.*, vol. 95, no. 6, pp. 3400–3408, June 1994.
- [186] S. D. Snyder and N. Tanaka, “Active control of vibration using a neural network”, *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 819–828, July 1995.
- [187] D. Pavisic, L. Blondel, and J. P. Draye, “Active noise control with dynamic recurrent neural networks”, *ESANN Proceedings, Brussels*, pp. 45–50, April 1995.
- [188] C. L. Nikias and M. Shao, *Signal processing with alpha stable distributions and applications*. Wiley, 1995.
- [189] R. Leahy, Z. Zhou, and Y. C. Hsu, “Adaptive filtering of stable processes for active attenuation of impulsive noise”, *ICASSP'95*, vol. 5, pp. 2983–2986, 1995.
- [190] E. R. Pike, J. G. McWhriter, M. Bertero, and C. de Mol, “Generalised information theory for inverse problems in signal processing”, *IEE Proceedings*, vol. 131, no. 6, pp. 660–667, October 1984. Part F.
- [191] G. E. Waranaka, L. A. Poole, and J. Tichy, “Active acoustic attenuator”, *US Patent no. 4473906*, 1984.
- [192] L. A. Poole, G. E. Waranaka, and R. C. Cutter, “The implementation of digital filters using a modified Widroff-Hoff algorithm for the adaptive cancellation of acoustic noise”, *ICASSP'84*, vol. 2, pp. 2171–2174, 1984.
- [193] A. O. Steinhardt, “Householder transforms in signal processing”, *IEEE ASSP Magazine*, pp. 4–12, July 1988.

- [194] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. Mc Graw Hill, 1991.
- [195] G. C. Goodwin and K. S. Sin, *Adaptive filtering, prediction and control*. Prentice Hall, 1984.
- [196] S. Grossmann and S. Thomas, “Invariant distributions and stationary correlation functions of one-dimensional discrete processes”, *Z. Naturforsch.*, vol. A32, pp. 1353–1363, 1977.
- [197] E. Kreyszig, *Advanced engineering mathematics*. Wiley, 1993.