Recursive Estimation of Driving-Forces from Nonlinear Nonstationary Systems with Unknown Dynamics

RECURSIVE ESTIMATION OF DRIVING-FORCES FROM NONLINEAR NONSTATIONARY SYSTEMS WITH UNKNOWN DYNAMICS

BY

ULAŞ GÜNTÜRKÜN, M.A.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2010) McMaster University
(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Recursive Estimation of Driving-Forces from Nonlinear Nonstationary Systems with Unknown Dynamics
AUTHOR:	Ulaş Güntürkün M.A.Sc., (Electronic and Communications Engineering) Yildiz Technical University, Istanbul, TURKEY
SUPERVISORS:	Dr. Hubert deBruin, Dr. James P. Reilly
NUMBER OF PAGES:	xix, 144

This dissertation is dedicated to my family for their continuous support throughout my studies, and to Mustafa Kemal ATATÜRK, the founder of secular Turkish Republic, for encouraging and making it possible for the citizens of his nation to recognize science as the only true guide in life.

"We do not consider our principles as dogmas contained in books said to come from heaven. We derive our inspiration, not from heaven, or the unseen world, but directly from life."

Mustafa Kemal ATATÜRK.

Abstract

We address a functional analysis-based method for the estimation of driving-forces from nonlinear dynamic systems in this thesis. The driving-forces account for the perturbation inputs or the irregular variations in the internal variables of a dynamic system. These inputs are hidden from the observer most of the time if not always. Reconstruction of such inputs when there is too little or no prior knowledge to build a mathematical model to describe the system's behavior is an important problem in many cases in physics and engineering. To this end, we propose a method for the recursive estimation of driving-forces without the availability of an analytic model of the unknown physical phenomenon.

The underlying idea of the proposed estimator is to predict the observables one-step ahead of the current time instant, and then retrieve the driving-force from the prediction error. This idea is embodied by predicting the observables using a bank of echo state networks (ESN) in an online fashion, extracting the raw estimates from the prediction error, and then finally smoothing these estimates in separate adaptive filtering stages. The approach described herein distinguishes itself from the similar methods in the literature in its adaptivity and its greater immunity against varying environmental uncertainties. The adaptive nature of the estimator enables us to retrieve both slowly and rapidly varying driving-forces accurately in presence of model or sensor noises, which are illustrated by experiments in the subsequent chapters of this thesis. In particular, some chaotic/stochastic nonlinear models are studied in controlled experiments. The estimation quality of the proposed approach is judged with a reference to the Posterior Cramer-Rao Lower Bound as a theoretical lower limit on the estimation error.

The Bayesian and Maximum-Likelihood (ML) methods are also studied for the estimation of driving-forces when partial or full information is available on the mathematical description of the unknown system. These methods serve as practical merits of assessment for the proposed driving-force estimator. Moreover, a direct performance comparison between the proposed estimator and a favorable estimation scheme of a similar kind is provided, which confirms the advantages of the proposed approach.

The proposed method is tested on a real-world application on the extraction of sun's magnetic flux from the sunspot time series. It is illustrated that the results obtained by the proposed estimator are in close agreement with the results of two other analytical studies.

Finally, a solution to a real problem in practice is proposed using the method. Specifically, extracting the signature of a small random target embedded in the sea surface is addressed using the live recorded data collected with the McMaster IPIX radar. This is the first specific realization of a radar scene analyzer for the cognitive radar reception in the literature to the author's best knowledge.

The material in this thesis is presented in a sandwich thesis format, combining two peer reviewed, published journal articles, and another journal article that is prepared for submission. An additional chapter that provides the background material is included for the completeness of the presentation.

Acknowledgements

Dr. J. Reilly, Dr. H. deBruin, Dr. T. Kirubarajan and Dr. M. Grasselli of McMaster University are greatly acknowledged for their suggestions and contributions to this work. The author would like to give special thanks to Dr. J. Reilly and Dr. H. deBruin for their support during the author's studies at McMaster University. Dr. H. Jaeger of Jacobs University Bremen is also acknowledged for providing a thorough review of the predictive modeling aspects of this work that was presented in a journal article, which was later withdrawn due to strict time constraints imposed by the Ph.D. comprehensive examination of the author. Many thanks are owed to Dr. Solanki, Dr. Schuessler and Dr. Fligge for permitting us to reproduce the corresponding curve in Fig.3.6. The author would like express his appreciation to two anonymous reviewers of the *Physica D* article (Güntürkün, 2010a), and three other anonymous reviewers of the *IEEE* article (Güntürkün, 2010b) and the associate editors of the respective articles for providing thorough reviews and for their invaluable suggestions and comments, which have greatly enriched the presentation of the material in this thesis. Special thanks are owed to the author's friends and colleagues, K. Atalay, P. Setoodeh, Y. Xue, F. Castiglione, J. Vincent, R. Abiprojo, O. Cakir, N. Costa, J. Bondy, and K. Winklund, who have made the time spent at McMaster University much more enjoyable. The author is especially grateful to Virginia and Oktay Aksan for being always very welcoming. Dr. S. Haykin of McMaster University is also acknowledged for his involvement in the early stages of this work.

Co–authorship

The writer of this thesis is the sole author of the two peer reviewed published journal articles (Güntürkün, 2010a,b), which are presented in Chapter 3. The article (Güntürkün, 2010a) has been published in the journal *Physica D: Nonlinear Phenomena*, which is an interdisciplinary physics journal widely recognized in the studies of nonlinear dynamic systems. The article (Güntürkün, 2010b) has been published in the special issue on *Non-Rayleigh Reverberation and Clutter* in the *IEEE Journal of Oceanic Engineering.*

Dr. James P. Reilly, Dr. Hubert deBruin and Dr. Thiagalingam Kirubarajan are the three co-authors of the article (Güntürkün *et al.*, 2010) that is prepared for submission to the *Signal Processing* journal and included in Chapter 4. The co-authors contributed to the work given in Chapter 4 by providing thorough advice on the analysis of the performance and the limitations of the proposed approach. The development and the software realizations of the methodologies, and planning and organization of the co-authored article (Güntürkün *et al.*, 2010) have been carried out by the author of the thesis.

Notation and abbreviations

Notations

u_n Driving-force (hidden input) at discrete-time n

x_n, $\mathbf{x}_n$ The scalar state variable, and the state vector respectively at discrete-time n

y_n The measured sample variable at discrete-time n

$g(.)$ The continuous and measurable mapping that describes the state transition

$h(.)$ The continuous and measurable mapping that relates the measurements to the state

$f(.)$ The mapping that governs the hybrid systems, which relates the previous measured sample to the current one

$\tilde{f}(.)$ An approximation to $f(.)$

ω Process (dynamic) noise

v Measurement noise

$\mathbb{R}$ The set of real numbers

$\mathbb{N}$ The set of natural numbers

$\mathbb{E}[.]$ Statistical expectation operation

σ_ω^2 The variance of the process noise

σ_v^2 The variance of the measurement noise

$\mathcal{T}$ Number of samples

T (Superscript) matrix transpose

$\mathbf{s}_n$ The state of the echo state network (neuronal state) at discrete-time n

$\mathbf{w}^{in}$ The vector of input weights for the echo state network

$\mathbf{W}$ The internal weight matrix for the echo state network

$\mathbf{w}^{out}$ The vector of output weights for the echo state network

Φ The time-averaged autocorrelation matrix of the neuronal state for the operation of RLS algorithm

$\mathbf{P}$ The inverse of the autocorrelation matrix for the neuronal state

λ_{RLS} The forgetting factor for the RLS algorithm

$\mathbf{k}$ The RLS gain vector

ς The connectivity rate of the dynamic reservoir of the echo state network

ρ The spectral radius of the internal weight matrix for the echo state network

N The number of neurons in the dynamic reservoir of the echo state network

V The number of echo state networks in the ESN bank

z_n The output variable the echo state network at discrete-time n

e_n The prediction error for the echo state network at discrete-time n

$\mathbf{r}_n$ The tap-input vector for the regularized adaptive estimator

$\hat{r}_n$ The output variable for the regularized adaptive estimator

α_n The prediction error for the regularized adaptive estimator

β_n The difference between the two consecutive outputs of the regularized adaptive estimator

L The number of tap-weights for the regularized adaptive estimator

J The cost function associated with the regularized adaptive estimator

J_s The standard error term for the definition of the cost function associated with the regularized adaptive estimator

J_{reg} The regularizing error term for the definition of the cost function associated with the regularized adaptive estimator

λ The regularization constant

$\mathbf{a}_n$ The tap-weight vector for the regularized adaptive estimator

μ_n The step-size parameter for the regularized adaptive estimator

$p(.)$ Probability density function

$\mathbb{P}$ Discrete probability

θ The hidden parameter

$\sigma_{\hat{u}}^2$ Time-averaged variance of estimation error

P_u The power of the driving-force

$\mathbf{D}_n$ Measurements in terms of spectral power up to time n

ξ_n^r The event of a target being present in resolution cell r at time n

$\sqrt{\tau_n}$ The square root texture component at time n

∇ The gradient operator

$\mathcal{C}(.)$ The number of floating-point operations

$\mathbf{I}$ The identity matrix

P The order of all-pole filter responsible for the generation of the driving-force

M The number of particles

tanh(.) The hyperbolic tangent function

Abbreviations

ADFE The Adaptive Driving-Force Estimator

AM Amplitude Modulation

BTT The Bayesian Target Tracker

DR Dynamic Reservoir

EM Expectation Maximization

EM-PF Expectation Maximization Particle Filter

ESN The Echo State Network

FLOP Floating-point operation

GMRF Gauss-Markov Random Field

IEEE The International Institute of Electrical and Electronic Engineers

IPIX Intelligent PIXel Processing Radar

KF Kalman filter

KS Kalman smoother

LLN Law of Large Numbers

LMS Least Mean Square

L-Map Logistic Map

ML Maximum Likelihood

MSE Mean Square Error

MR-Map Moran-Ricker Map

NMSE Normalized Mean Square Error

PF Particle Filter

PDF Probability Density Function

RBF Radial Basis Function

RC Reservoir Computation

RLS Recursive Least Squares

PCRB Posterior Cramer-Rao Lower Bound

RBPF Rao-Blackwellized Particle Filter

RMLP Recurrent Multilayer Perceptron

RNN Recurrent Neural Network

RSA Radar Scene Analysis

SNR Signal-to-Noise Ratio

SPF Standard Particle Filter

STFT Short-Term Fourier Transform

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction and Problem Statement

1.1 Introduction

Many real-life dynamic systems generate nonstationary outputs due to the inevitable presence of measurement noise, dynamic noise and changes in the internal system parameters or environmental conditions in the course of time. Measurement noise implies stochasticity, whereas the perturbations may cause the system parameters to be time-varying, and the combined effects of these factors result in a nonstationary process arising from a perturbed dynamic system. The perturbation signals (driving-forces) can be viewed as unknown inputs to the unperturbed dynamics. They are hidden from the observer most of the time if not always, and estimating them can be of practical interest in many cases in physics and engineering.

1.1.1 Some Examples of Perturbed Dynamic Systems

Here we address some examples of how in many physical phenomena we have a dynamic system driven by a force that is not observable:

Sunspot Time Series: A sunspot is a cooler darker spot appearing on the sun's photosphere whose mechanism is not exactly known. In 1848, Rudolph Wolf devised a daily method of estimating solar activity by counting the number of individual spots and groups of spots on the face of the sun (NGDC, 2006). The variation in the sun's magnetic flux plays the role of a perturbation process impacting the evolution of the sunspot numbers' time series. Solanki *et al.* (2002) developed a mathematical model to extract the information pertaining to sun's magnetic flux.

Sea Clutter: Sea clutter refers to the radar backscatter from an ocean surface. The study of sea clutter is not only of theoretical importance but also of practical importance because it places severe limitations on the detectability of point targets (e.g., low-flying aircraft, small marine vessels, navigation buoys, and small pieces of ice) on or near the sea surface. The two fundamental types of waves (i.e. gravity and capillary waves) dictate the roughness of the sea surface, and govern the clutter dynamics (Haykin *et al.*, 2002). When a small target is embedded in the sea surface, the movement of the target will be also dominated by the governing sea waves. Then a small target can be modeled as an additional random perturbation in the sea surface, whose dynamics are closely coupled by those of the sea. Hence the estimation of a small random target's signature can be cast as a driving-force estimation problem within the radar scene analysis.

Wireless Communication Channel: In a general wireless communications scenario, the transmitted signal reaches the receiver by different transmission paths with different path lengths. This causes the received signal to be a superposition of different reflected signals, which have varying delays due to their path lengths. Since the receiver could be moving, fast fading occurs and the channel is therefore time-varying. In such situations, the physical and geometrical properties of the environment (buildings, trees, etc.) and movement of the receiver play the role of perturbation inputs.

Physiological Processes: In living organisms, heart activity, breathing, muscle tremor and voice production are some examples of physiological processes. All these processes have some degree of nonstationarity due to the perturbations coming both from the environment and from different systems of the vital activity of the organism.

Seismic Data: Earth's deep interior has been largely influenced by earthquakes, most of which are caused by the sudden movement of rock masses along a fault. As these rocks grind together, energy is released and vibrations are produced, which we call seismic waves. The speed of seismic waves in rocks depends on several environmental variables, the most important of which are pressure and temperature (DGPS, 2010). Therefore, seismic waves, being the reflection of the rock motions, are perturbed by those driving-forces.

1.1.2 Problem Statement

In light of these physical examples, we can identify three estimation scenarios:

1. The state and driving-force are both one-dimensional, as in the case of sunspot data.

2. The state is multidimensional, and the driving-force is common to every element of the state, as in the case of sea clutter, where a population of the local

scatterers are perturbed by the same small target. The target would play the role of a driving-force.

3. The state and driving-force are both multidimensional; it is in such a situation where we need prior information on the evolution of state to estimate the driving-forces.

The material covered in this thesis is applicable to scenarios 1 and 2.

Consider a one-dimensional (possibly nonlinear) dynamic system as described in (1.1), whose state-space model consists of two parts:

- A process equation that describes the evolution of state under the action of process noise and a driving-force; the force is unknown.
- A measurement equation that defines the observables buried in measurement noise.

These two equations can be represented respectively as

$$\begin{aligned} x_{n+1} &= g\left(x_n, u_n\right) + \omega_{n+1} \qquad (1.1) \\ y_{n+1} &= h(x_{n+1}) + \upsilon_{n+1}. \end{aligned}$$

The continuous and measurable mapping $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ defines the state transition, and $h : \mathbb{R} \rightarrow \mathbb{R}$ defines the evolution of observables respectively. $x_n \in \mathbb{R}$ denotes the state, $u_n \in \mathbb{R}$ is the unknown input signal (driving-force), $y_n \in \mathbb{R}$ denotes the observable, $\omega_n \in \mathbb{R}$ is the dynamic noise, and $\upsilon_n \in \mathbb{R}$ is the measurement noise, all at discrete time $n \in [0, \mathcal{T}]$. $\mathcal{T} \in \mathbb{N}$ denotes the number of data samples and $\mathbb{R}$ denotes the real space. Note that (1.1) can be extended to cover the cases where the state is multidimensional without loss of generality.

In general terms, the problem that we are posing is similar to the state-estimation problem, where the requirement is to estimate the state x_n. Our problem, however, is different: We wish to estimate the driving-force u_n without knowledge of $g(.)$ and $h(.)$ given that $h(.)$ is invertible.

1.2 Literature Review

The hidden input estimation problem has received considerable attention in both the physics and engineering disciplines. We can divide the numerous contributions in the literature to two main categories.

In the first category, an analytical description of the system model is readily available. The second category covers such cases where the system model is unknown. It is these cases that we address in this thesis. We present below a summary of the main contributions in both categories.

1.2.1 Input Estimation in Fully Identified Systems

Linear Deterministic Systems

The system under study is represented by a linear state-space model without any uncertainties. Mainly, linear algebra techniques are used for inverting the system, and estimating the unknown input. Such an estimator is usually referred to as an 'observer' in the automatic control literature (Guan and Saif, 1991), (Darouach *et al.*, 1994), (Hou and Patton, 1998a). Bhattacharyya (1978) takes a geometrical approach as an alternative method. Singular value decomposition (SVD) is presented by Fairman *et al.* (1984) for reconstruction of the hidden input.

Linear Stochastic Systems

The system model is still linear, but includes additive noise to represent the uncertainties about the system dynamics and measurement errors. Most of the literature for the study of such cases is based on the generalization of the linear optimum Gaussian filtering (i.e. Kalman filter). The generalized Kalman filter (Gillijns and Moor, 2006) is realized as two interconnected steps, hence the estimation of state and input are linked. Hou and Patton (1998b) decouple these two operations, exploiting the innovations approach in Kalman filtering. deconvolution approach is studied by Nicolao *et al.* (1997) based on a linear system with a known impulse response. The solution is based on regularized least squares estimation for physiological processes.

Nonlinear Deterministic Systems

In these cases, a commonly employed approach is to transform the nonlinear dynamics to a new set of coordinates where linear methods can be applied, (Krener and Respondek, 1985), (Xia and Gao, 1989). Another approach that is widely practiced is applicable to nonlinearities of the Lipschitz kind, which allows local linear approximations. Zhu and Han (2002) linearize the estimation error using the Lipschitz condition. Then based on the stability of the estimated error, a Riccati equation is solved and the estimates are obtained. Boutayeb *et al.* (2002) determine the (linearized) model matrices given the Lipschitz nonlinearity, and estimate hidden input using linear algebra techniques. Arcak and Kokotovic (2001) exchange the Lipschitz restriction by the circle criterion (a theorem showing stability in nonlinear systems). The circle criterion is evaluated by LMI (linear matrix inversion) computations. In an article by Ha and Trinh (2004), the previous approaches are generalized to nonlinearities which are not necessarily Lipschitz any more, but rather are state-dependent nonlinearities. The estimation method depends on the solution of the Riccati equation by the LMI. The state and input are simultaneously estimated.

Nonlinear Stochastic Systems

Pillonetto and Saccomani (2006) study a nonlinear system where the state equation is noise-free, and the stochasticity is limited to measurement uncertainties. A Kalman smoother is used for suppressing the measurement noise. Then differential algebra techniques are used for estimating the input from the resulting supposedly deterministic system. In an article by Pillonetto and Bell (2004), the hidden parameter is buried in both the state and measurement equations. A Laplacian approximation is used for representing the likelihood of the unknown parameter, and maximum likelihood (ML) estimates are obtained to approximate the exact system model.

A more interesting input estimation scheme in nonlinear stochastic systems is arguably the application of Bayesian methods. The literature on the Bayesian input estimation however is limited to the estimation of static variables. In another work by Pillonetto and Bell (2007), results of (Pillonetto and Bell, 2004) are generalized by assuming that the prior pdf on the unknown parameters is known. Then a Bayesian solution is studied. The expectation maximization (EM) algorithm (Dempster *et al.*, 1977) combined with a Kalman smoother is shown to be an effective method to estimate the fixed parameters. Moon (1996) mention some general applications of EM.

Andrieu and Doucet (2003) derive an online version of EM. In two more recent papers (Zia *et al.*, 2007, 2008), the measurement model is assumed unknown, and it is estimated by virtue of estimating the parameters of a mixture of Gaussians (MoG) model. A different approach is augmenting the state model for the estimation of hidden parameters. Kitagawa (1998), estimate the unknown scalars of the system equations and noise variances by this approach based on sequential Markov Chain Monte Carlo methods (MCMC) (or particle filters). Higuchi (1997) replace the prediction step in particle filters by some elements of genetic algorithms, namely crossover and mutation. Storvik (2002) assumes that the distribution of the hidden parameter depends on the state vector through some low-dimensional sufficient statistics, and the hidden parameter is marginalized from the posterior. Liu and West (2001) fit a random walk model to the evolution of the hidden static parameters and they introduce kernel smoothing to compensate the artificial dynamic behavior of the random walk model. Zhang *et al.* (2006) modify a particle filtering algorithm for the estimation of unknown input variable. Specifically, the state transition operation is performed without the unknown input, and the value of the unknown input variable is chosen from a known constellation that maximizes the measurement likelihood. This operation is inserted as an intermediate step between prediction and update steps of the particle filter. Zhang *et al.* (2007) apply the same idea for the estimation of a static parameter.

1.2.2 Input Estimation in Unknown Systems

This scenario has received attention mostly from the physics literature, where input estimation is termed "driving-force estimation", or "perturbation estimation" in the relevant papers. In such problems, lack of sufficient prior knowledge of the system prompted researchers to resort to blind approaches based on some weaker assumptions. The earliest work for the estimation of driving-forces from systems with unknown dynamics can be dated back to the recurrent plots (Eckmann *et al.*, 1987). The idea behind recurrence plots is to construct windows of data with a suitable embedding length, and pass the scaled norms of the moving distance between the windows through a kernel for varying lags. The kernel is usually the Heaviside step function. The driving-force information is searched within these processed distances between the windows. Hence, the recurrent plots may be considered as the earliest realization of the prediction error idea. Casdagli (1997) revisited the recurrent plots by introducing an averaging technique for deterministic systems. Then the driving-force can be reconstructed up to an arbitrary amplitude transformation. An improved approach is presented by Schreiber (1997, 1999, sec. 4.2). Two segments from the time series are constructed. One segment is used for predicting another segment. The driving-force is searched in this cross-prediction error. As argued by the authors themselves, a strong limitation of this approach is the following: The duration of overlap between two consecutive segments, and the length of each segment must be chosen properly so as to guarantee that one segment contains enough information to predict another segment.

Verdes *et al.* (2001) solidified the prediction error idea using a radial basis function (RBF) network for predicting the overlapping intervals of time series. The average prediction errors are calculated within these intervals, and the time-average of the driving-force is reconstructed from the prediction error subject to an arbitrary amplitude transformation. As proposed by Casdagli (1997); Eckmann *et al.* (1987); Schreiber (1997, 1999), the method is also applicable only when the driving-force is known to be varying very slowly. However, the authors show their method to be more effective than that of Schreiber (1997, 1999) thanks to accurate modeling by an RBF network. While the method of Schreiber (1997, 1999) is relatively insensitive to the number of overlapping samples, the performance of Verdes *et al.* (2001) heavily depends on the number of samples that overlap between two consecutive intervals.

Szeliga *et al.* (2002, 2003a,b) further improved the approach presented in (Verdes *et al.*, 2001). Specifically, a feedforward multilayer perceptron (MLP) is used for predictive modeling, to which an extra input node is attached to feed the driving-force estimates into the network. Then the MLP is trained with and without the extra input. The combined error of two training phases is minimized by the gradient descent. Again, the time-average of the driving-force is reconstructed in consecutive intervals subject to an unknown scale transformation. Verdes *et al.* (2004, 2006) showed that feeding the driving-force estimates into the neural network enables modeling of nonstationary time series in a more accurate fashion.

Consequently, we can summarize the relevant literature as follows: All the contributions by Casdagli (1997); Eckmann *et al.* (1987); Schreiber (1997, 1999); Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006) can be considered as different realizations of reconstructing very slowly varying driving-forces from the prediction error in an offline fashion. The work presented by Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006) differs from those of Casdagli (1997); Eckmann *et al.* (1987); Schreiber (1997, 1999) in that the prediction of the observables is handled in a more principled manner using RBFs and MLPs. As a result, Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006) have reported a superior performance compared to the preceding techniques.

Batch processing [1] is common to all the contributions by Casdagli (1997); Eckmann *et al.* (1987); Schreiber (1997, 1999); Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006) with the objective of retrieving slowly varying forces. In many real-world applications however (e.g. sea clutter, wireless channel), it is essential to estimate the signal of interest in an online fashion and update these estimates as the new data become available. Also, the driving-signal may not change very slowly in many real-world problems. Motivated by these facts, and building on the contributions listed above (mainly on the work of Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006)), we described an adaptive sequential method for driving-force estimation in two peer reviewed, published journal articles, (Güntürkün, 2010a,b).

We applied the proposed driving-force estimator (Güntürkün, 2010a) to the estimation of both slowly and rapidly varying forces. Also, following from the work of Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006), we demonstrated the algorithm's accuracy on a real world application, where the objective is to estimate the sun's magnetic flux using the sunspot time series. We have also provided a comparison between the proposed method and that of Verdes *et al.* (2006), which revealed that the proposed method provides a much greater immunity against the effect of additive noise.

We proposed a solution to the radar scene analysis (RSA) problem for the cognitive radar receiver (Güntürkün, 2010b) using the proposed approach. Building on the cognitive radar reception idea of Haykin (2006), we showed that the signatures of small, random targets can be retrieved using the proposed driving-force estimator exploiting the texture modeling of the sea clutter. We demonstrated the results of

[1] In the batch processing or offline processing, all the available data are gathered first, then presented to the processor (e.g. the neural network) at successive times before retrieving the quantity of interest.

the algorithm qualitatively on live-recorded radar returns collected with the McMaster IPIX radar.

1.3 Thesis Organization

This thesis is organized in sandwich thesis format in accordance with the regulations of the School of Graduate Studies (SGS) at McMaster University. The presentation in the thesis is built around the published journal articles (Güntürkün, 2010a,b), and another article, (Güntürkün *et al.*, 2010) that is prepared for submission.

In Chapter 2, the background material is presented. In particular, the online predictive modeling with the echo state networks (ESNs) has been described. Modeling of the unknown dynamic environment with the ESNs is the basis on which the proposed driving-force estimator is built. The driving-force estimation problem is also handled using the Bayesian techniques when knowledge of the underlying physical system is partially or fully available. Detailed descriptions of the Bayesian methods are also given in Chapter 2.

The proposed driving-force estimator is described in detail in Chapter 3 with applications to chaotic dynamic systems and the sunspot time series. A solution is proposed to the radar scene analysis (RSA) problem using the proposed method also in Chapter 3. The material in Chapter 3 is mostly based on the contributions of the two articles (Güntürkün, 2010a,b).

The results for the application of the Bayesian inference to the driving-force estimation problem are illustrated and compared with the proposed method in Chapter 4. All driving-force estimation algorithms are provided an equivalent amount of computational resources for a fair comparison. The computational complexities are calculated in terms of the floating point operations (*flops*), which provide an exact implementation cost of the respective algorithms.

Chapter 5 concludes the thesis with a summary of the contributions of the work to general knowledge and remarks on future research.

All the references are cited in author-year format, which is different than the IEEE numerical format in the original articles.

At the beginning of each chapter, the reader is alerted for possible overlaps between some material that may appear in different parts of the thesis. The occurrences of such overlaps are kept as rare as possible throughout the thesis.

Chapter 2

Background

This chapter is organized to provide background material for the driving-force estimation techniques that are presented in Chapter 3 and Chapter 4. The first section of this chapter is devoted to the description of the echo state networks (ESNs), which are used for the online predictive modeling of the unknown environment. ESNs constitute the basis of the proposed adaptive driving-force estimator. The other operations encompassed by the proposed algorithm are throughly analyzed in Chapter 4, and are hence omitted herein. The remaining sections of this chapter provide background on the Bayesian techniques, which are also applied to the driving-force estimation problem in Chapter 4 for comparison with the proposed method.

2.1 Recursive Predictive Modeling with Echo State Networks

The prediction of the time series measured from the unknown environment is one of the key operations embodying the proposed adaptive driving-force estimator. Echo state networks (ESNs) are used for predicting the observables one-step into the future. Online or one-step prediction implies that the current observable of the time series y_n is used for predicting the new observable, y_{n+1}.

The ESN is a recent class of the recurrent neural networks (RNN) (Jaeger, 2001, 2003; Jaeger and Haas, 2004), whose principles are based on the idea of *reservoir computation* (RC). The basic idea behind the RC is to construct a dynamic reservoir (DR) that is made up of randomly and sparsely connected recurrent neurons with fixed internal weights, and train only the linear readout layer (i.e. the weights $\mathbf{w}_n^{out,(i)}$ as shown in Fig.2.1). Both the ESN and its biologically inspired counterpart liquid state machine (LSM) are shown to perform well on a variety of different tasks (Jaeger, 2001, 2003; Natschlager *et al.*, 2002). The ESN distinguishes itself from other popular recurrent neural networks (e.g. recurrent multilayer perceptrons (RMLP), real time recurrent learners (RTRL)) in its simple and effortless training, built-in short-term memory and adequacy for real-time processing.

We consider V ESNs configured as in Fig.2.1 running in parallel, operating on the same input-output data pair at each instant n, which we refer to as the 'ESN bank' that will be elaborated in Section 2.1.1. The input-output relation for one ESN

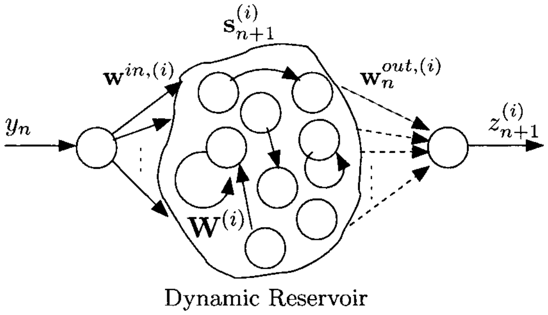


Figure 2.1: Echo State Network (ESN) with single input, single output. The number of recurrent units in the DR is denoted by N. The trainable output weights, $\mathbf{w}_n^{out,(i)}$ are represented with the dotted lines. The superscript (i) denotes the ith ESN in the bank, $i = 1, 2, \ldots, V$.

configuration can be expressed as

$$\begin{aligned} \mathbf{s}_{n+1}^{(i)} &= \tanh\left(\mathbf{W}^{(i)}\mathbf{s}_n^{(i)} + \mathbf{w}^{in,(i)} y_n + \phi_{n+1}^{(i)}\right) \\ z_{n+1}^{(i)} &= \left(\mathbf{w}_n^{out,(i)}\right)^T \mathbf{s}_{n+1}^{(i)} \end{aligned} \tag{2.2}$$

where $\mathbf{s}_{n+1}^{(i)}$ is the $N \times 1$ state vector of the hidden units for the ith ESN at time $n+1$, $\mathbf{w}^{in,(i)}$ denotes the $N \times 1$ vector of input weights, and $\mathbf{w}_{n+1}^{out,(i)}$ is the $N \times 1$ vector of output weights at time $n+1$. The superscript ^{T} denotes the matrix transpose. $\phi_{n+1}^{(i)}$ is the $N \times 1$ artificial noise sequence inserted for enhancing the numerical stability of the weight adaptation algorithm. Internal connections (denoted by the $N \times N$ matrix $\mathbf{W}^{(i)}$) within the DR and the input weights remain constant. This way, training of the ESN boils down to a linear regression task. The $N \times 1$ output weights $\mathbf{w}^{out,(i)} = [w_1^{out,(i)}, ..., w_N^{out,(i)}]^T$ are trained using the recursive least squares (RLS) algorithm.

The RLS algorithm is briefly given as follows,

$$\begin{aligned}\mathbf{k}_{n+1}^{(i)} &= \frac{\lambda_{RLS}^{-1}\mathbf{P}_n^{(i)}\mathbf{s}_{n+1}^{(i)}}{1+\lambda_{RLS}^{-1}\left(\mathbf{s}_{n+1}^{(i)}\right)^T\mathbf{P}_n^{(i)}\mathbf{s}_{n+1}^{(i)}} \\ \mathbf{w}_{n+1}^{out,(i)} &= \mathbf{w}_n^{out,(i)} + \mathbf{k}_{n+1}^{(i)}e_{n+1}^{(i)} \\ \mathbf{P}_{n+1}^{(i)} &= \lambda_{RLS}^{-1}\mathbf{P}_n^{(i)} - \lambda_{RLS}^{-1}\mathbf{k}_{n+1}^{(i)}\left(\mathbf{s}_{n+1}^{(i)}\right)^T\mathbf{P}_n^{(i)}\end{aligned} \tag{2.3}$$

where $\mathbf{k}^{(i)}$ is $N \times 1$ gain vector, and λ_{RLS} is the forgetting scalar associated with the (i)th ESN in the bank. The $N \times N$ matrix $\mathbf{P}_n^{(i)}$ denotes the inverse of the time-averaged correlation matrix of the neuronal state vector $\mathbf{s}_n^{(i)}$, which is given as

$$\Phi_n^{(i)} = \sum_{n=1}^{T} \left(\lambda_{RLS}\right)^{T-n} \mathbf{s}_n^{(i)} \left(\mathbf{s}_n^{(i)}\right)^T, \tag{2.4}$$

and $\mathbf{P}_n^{(i)} \triangleq \left(\Phi_n^{(i)}\right)^{-1}$.

Success of the ESN design is based on how well the DR is constructed. The DR setup is determined by the tuning the following free parameters: Connectivity rate (ς)—(a measure of sparseness of the internal weight matrix, $\mathbf{W}$), spectral radius (i.e. the largest eigenvalue) (ρ) of $\mathbf{W}$ matrix—(a measure of internal time-scale), and the number of neurons (N). The 'echo state property' refers to the fact that the initial effects "wash out" in the course of time. After the transition period is completed, the network output is a function of solely the trainer input history. Hence the neural network provides a linear combination of the echos of the trainer signal. For this property to hold, $0 < \rho < 1$ is required. This property also guarantees the asymptotic stability of the ESN dynamic reservoir as long as the network's own output $z_{n+1}^{(i)}$ is not fed back into the DR (Jaeger, 2001, p.8, proposition 3). For the construction of the internal weight matrix, we follow the echo state recipe in (Jaeger, 2001), which is reproduced below:

1. randomly generate an internal weight matrix, $\mathbf{W}_0$,
2. normalize the initial matrix $\mathbf{W}_0$ to another matrix, $\mathbf{W}_1$, such that $\mathbf{W}_1 = (1/\left|\rho_0\right|)\mathbf{W}_0$, where ρ_0 is the spectral radius of (i.e. the largest eigenvalue) $\mathbf{W}_0$,
3. assign the desired spectral radius to the internal weight matrix by setting $\mathbf{W} = \rho\mathbf{W}_1$, hence the matrix $\mathbf{W}$ has the spectral radius $\rho < 1$,
4. the untrained network $(\mathbf{w}^{in}, \mathbf{W})$ has the echo state property regardless of how $\mathbf{w}^{in,(i)}$ is constructed.

The entries of input weight vector $\mathbf{w}^{in}$ are drawn from the uniform distribution in the interval $[-1, 1]$.

The echo state property is illustrated on a simulation experiment, whose results are depicted in Fig.2.2 for a single ESN. For the experimental setup, 30 samples from the logistic map dataset are drawn. In (2.5), we describe the experimental data, where the state equation defines the logistic map,

$$\begin{aligned} x_{n+1} &= r_o x_n \left(1 - x_n\right) \\ y_{n+1} &= x_{n+1} + v_{n+1}. \end{aligned} \tag{2.5}$$

r_o is the bifurcation constant, which determines the region of operation for the logistic map. $r_o = 3.8$ is set so as to keep the map in the interesting chaotic regime. v_n is additive white Gaussian noise (AWGN), whose variance is set to $\sigma_v^2 = 0.5$. The initial state, $x_0 = 0.42$ is set. The logistic map is studied in greater detail in Section (3.1.4).

The task of the ESN to predict y_{n+1} using only the current sample y_n at its input. After the initial period (e.g. first 10 steps) is completed, the distinct echos of the teacher input appear the hidden layer (DR) as illustrated in the left-hand side of Fig.2.2. These echos, in a successfully constructed DR, compose a rich set of basis functions to approximate the desired response. The role of the output layer is then to optimally combine these echos to predict the desired response. As observed in right-hand side of Fig.2.2, the weighted linear combination of the echo functions follow the original observables closely.

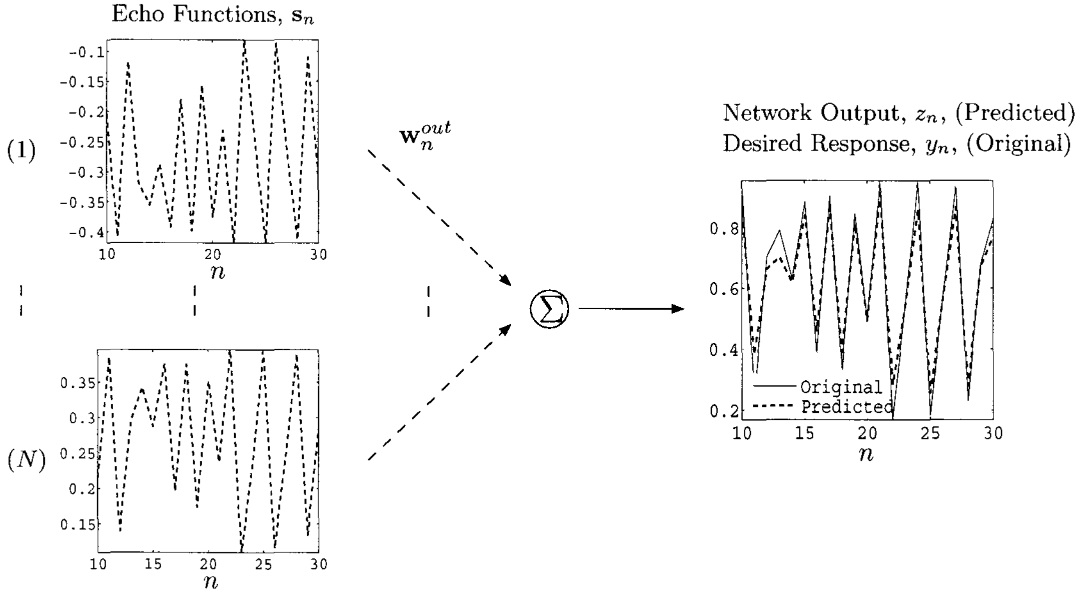


Figure 2.2: Online predictive modeling with echo state network. Left-hand side of the figure illustrates the echo functions sampled from a single ESN, (i.e. $V = 1$). The hidden layer signals (i.e. the echo functions) for the first and the Nth neuron are shown on the left with $N = 10$. The original time series and ESN approximation are shown on the right-hand side.

2.1.1 The ESN Bank Approach

Our experiments show that using a bank of ESNs rather than a single ESN enables approximation of the unknown dynamics in a much more accurate way. The reason for this is two fold: First; the fact that each ESN in the bank has a randomly connected DR assures that the ESNs will produce non-identical outputs despite the fact that they all receive the same teacher input. Therefore, V independent networks provide an ensemble averaging mechanism, which offers a greater immunity against the noise. Second, V decoupled DRs provide a much richer basis for approximating the nonlinear dynamics of the teacher input. Another advantage of the ESN bank approach is that the one-step mean square prediction error averaged over all ESNs is found to be fairly insensitive to the selection of the free parameters of the ESN.

We demonstrate the efficiency of the ESN bank approach by comparing an ESN bank with a 7 times larger single ESN of the same complexity. The numerical results herein are obtained by applying the proposed driving-force estimator to the estimation of the hidden input from the nonlinear stochastic system studied in detail in Section 4.2. Following from the complexity expression in (4.3), we can realize the adaptive driving-force estimator (ADFE) given a fixed complexity by setting

- $\{V = 1, N = 70\}$,
- or $\{V = 34, N = 10\}$ for a fixed ς.

The computational complexity is specified as the number of floating point operations (i.e. *flops*). The former configuration points to a single ESN and is realized at approximately 37.10^3 flops. The latter configuration is an ESN bank with an equivalent complexity. For both cases, we test the sensitivity of the ADFE to the selection of (ς)

and (ρ). Following the guidelines given by Jaeger (2001), we propose some candidate values for the free parameters such that $\varsigma \in \{0.3,\ 0.4,\ 0.5\}$ and $\rho \in \{0.1,\ 0.4,\ 0.7\}$, and measure the corresponding variance of the estimation error.

In Table 2.1, the variance of the estimation error is given vs. the varying values of ς and ρ. Clearly, the performance of the estimator is sufficiently insensitive to the selection of ESN free parameters. Although all 3^2 combinations of the pair (ς, ρ) are tried out, we only present the results for $\varsigma \in \{0.3,\ 0.4,\ 0.5\ |\rho = 0.1\}$ and $\rho \in \{0.1,\ 0.4,\ 0.7\ |\varsigma = 0.1\}$ for the sake of brevity. The remaining combinations do not lead to any significant changes from the results presented in Table 2.1.

In Table 2.2, the sensitivity of the ADFE to the selection of ESN parameters is analyzed if a single ESN were used rather than an ESN bank. It is illustrated that a single ESN still provides a reliable estimation performance suggested by its insensitivity to the selection of ς and ρ. However, comparing the rightmost columns of Table 2.1 and Table 2.2, it is clear that the ESN bank approach leads to a significant reduction in the variance of the estimation error compared to a single ESN of the same complexity.

Table 2.1: Sensitivity of the ADFE to the selection of the free parameters of an ESN bank for $N = 10$, $V = 34$. $\sigma^2_{\hat{u}_n} = 1/T \sum_{n=1}^{T} \mathbb{E}\left[(\hat{u}_n - u_n)^2\right]$ denotes the time-averaged variance of the estimation error.

ς	$\sigma^2_{\hat{u}_n}$	ρ	$\sigma^2_{\hat{u}_n}$
0.3	0.37	0.1	0.37
0.4	0.36	0.4	0.36
0.5	0.37	0.7	0.37

Table 2.2: Sensitivity of the ADFE to the selection of the free parameters of a single ESN for $N = 70$, $V = 1$. $\sigma^2_{\hat{u}_n} = 1/\mathcal{T}\sum_{n=1}^{\mathcal{T}} \mathbb{E}\left[(\hat{u}_n - u_n)^2\right]$ denotes the time-averaged variance of the estimation error.

ς	$\sigma^2_{\hat{u}_n}$	ρ	$\sigma^2_{\hat{u}_n}$
0.3	0.54	0.1	0.54
0.4	0.54	0.4	0.54
0.5	0.52	0.7	0.55

2.2 Bayesian Inference for Recursive Driving-Force Estimation

In this section, we provide an overview of the simulation-based Bayesian methods for the estimation of time-varying states of nonlinear dynamic systems. The material in this section is intended as a basis for more advanced particle methods that are used for the driving-force estimation in Chapter 4.

The Bayesian methods can be used for filtering, prediction or smoothing of the unknown states. Typically, the observer is provided only with a series of measurements, $\{y_n | n = 1, 2, \ldots, \mathcal{T}\}$, and the requirement is to make decisions about the hidden states, $\mathbf{x}_n$ in (2.6) without having a direct access to $\mathbf{x}_n$. An accurate mathematical description of the underlying dynamic system as given in (2.6) is also a crucial piece of information required for successful operations of the Bayesian methods. Let (2.6) represent the state-space description of the underlying system that is available to the observer,

$$\begin{aligned} \mathbf{x}_{n+1} &= g\left(\mathbf{x}_n\right) + \omega_{n+1} \qquad (2.6) \\ y_{n+1} &= h(\mathbf{x}_{n+1}) + v_{n+1} \end{aligned}$$

which differs from the description of the unknown system given in (1.1) in that we have incorporated the driving-force u_n in the augmented state, $\mathbf{x}_n$ in (2.6) for the ease of exposition.

Our focus in our applications is the online or sequential filtering problem. In sequential filtering, a probabilistic belief on the unknown state $\mathbf{x}_n$ is formed based on the history of the observables $\{y_1, \ldots, y_{n-1}\}$, and the knowledge of a model as in (2.6). This belief is represented by a probability density function (pdf), from which all the moments of the unknown state $\mathbf{x}_n$ can be calculated. Then at the arrival of each new data sample y_n, the belief on the unknown state is updated. In more formal terms, our problem is to estimate the filtering posterior probability density of the hidden states, $p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$. The vector $\mathbf{y}_{1:n}$ denotes the history of the observables, which is defined as $\mathbf{y}_{1:n} \triangleq \{y_1, \ldots, y_n\}$.

For a mathematical treatment of the Bayesian sequential filtering, let us assume that $p\left(\mathbf{x}_0|\mathbf{y}_0\right) \equiv p(x_0)$ is known a priori. Then the underlying idea of the sequential Bayesian estimation is to infer the pdf $p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$ in two steps in an iterative manner: prediction, and update.

Let us suppose that the pdf $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{n-1}\right)$ is available at time $n-1$. In the prediction stage, the prediction density $p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)$ is obtained by virtue of the Chapman-Kolmogorov equation as given in (2.7),

$$p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right) = \int p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right) p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right) d\mathbf{x}_{n-1} \tag{2.7}$$

where $p\left(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{1:n-1}\right) = p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right)$ has been made use of for the first factor in the integrand in (2.7). Note that this follows from the fact that the first line of (2.6) is a first-order Markov process. The transition prior density $p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right)$ is specified by the state transition model in (2.6) and the known statistics of ω_n.

At the instant n, the new measurement y_n becomes available, and it is used for updating the prediction density in (2.7) via Bayes' rule to obtain the desired posterior density,

$$p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right) = \frac{p\left(y_n|\mathbf{x}_n\right)p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)}{p\left(y_n|\mathbf{y}_{1:n-1}\right)}. \tag{2.8}$$

The normalizing constant $p\left(y_n|\mathbf{y}_{1:n-1}\right) = \int p\left(y_n|\mathbf{x}_n\right)p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)d\mathbf{x}_n$ depends on the likelihood function $p\left(y_n|\mathbf{x}_n\right)$, which is defined by the measurement equation in the second line of (2.6) and the known statistics of υ_n.

The optimum Bayesian solution given in predict/update equations in (2.7) and (2.8) constitute only a conceptual solution, and can not usually be determined analytically due to the possibly intractable integrals. In the sequel, we describe how the particle filters provide a suboptimum solution to this problem.

2.2.1 Sampling Importance Resampling (Standard) Particle Filter

The particle filters provide simulation based-filtering solutions that are used for approximating the posterior density functions of the unknown states by means of a weighted sum of discrete support points. These support points are called the *particles.* Then the desired estimates of the unobserved dynamic states are obtained by simulating and propagating these discrete pdf approximations via Bayes' rule. The key idea in the particle filtering is represented in (2.9),

$$p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right) \approx \sum_{i=1}^{M} w_{n-1}^{(i)}\delta\left(\mathbf{x}_{n-1} - \mathbf{x}_{n-1}^{(i)}\right) \tag{2.9}$$

where the posterior density $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$ at time $n-1$ is approximated by a weighted sum of particles, $\{\mathbf{x}_{n-1}^{(i)}|i=1,\ldots,M\}$. $w_{n-1}^{(i)}$ is the weight of the ith particle, and $\delta(.)$ is the delta function. The weights are normalized such that $\sum_{i=1}^{M} w_{n-1}^{(i)} = 1$.

Note that if we could sample directly from the true posterior, then we could have drawn the particle cloud $\{\mathbf{x}_{n-1}^{(i)}|i=1,\ldots,M\}$ according to the pdf $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$, and all the weights would have been set to $1/M$. This is known as the *uniform sampling*. However, in general, the posterior density $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$ can be very hard-to-sample from, and it can be available only for evaluation up to a normalizing constant. In such cases, we take an indirect approach to realize the approximation in (2.9). In particular, we resort to an auxiliary pdf, from which it is easy-to-sample. This intermediate pdf is called the *proposal density* or the *importance function*, and this approach is known as *importance sampling*. An illustration of the importance sampling idea is given in Fig.2.3, where the true posterior density is represented as a "weirdly-skewed" pdf, such that sampling directly from the true posterior is illustrated not to be feasible. In the same graph, the proposal density is presented with a more regular shape to represent a pdf, which is easy-to-sample from.

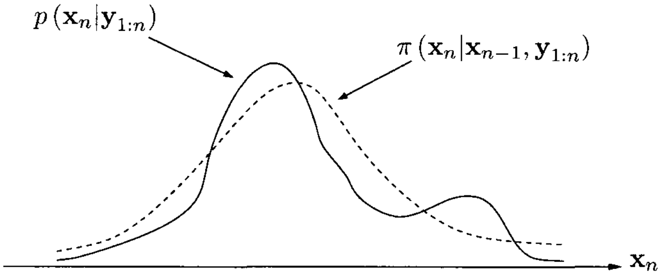

$p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$
$\pi\left(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_{1:n}\right)$
$\mathbf{x}_n$

Figure 2.3: Illustration of the importance sampling: the solid curve represents the true posterior density, whereas the dashed curve stands for the proposal density (importance function).

To elaborate on the importance sampling idea, Let $\pi\left(\mathbf{x}_{n-1}|\mathbf{x}_{n-2},\mathbf{y}_{1:n-1}\right)$ denote an importance function at $n-1$. Sampling of the particles from this importance function is represented by

$$\{\mathbf{x}_{n-1}^{(i)} | i = 1, \ldots, M\} \sim \pi\left(\mathbf{x}_{n-1} | \mathbf{x}_{n-2}, \mathbf{y}_{1:n-1}\right). \tag{2.10}$$

The importance weights associated with the particles in (2.10) are determined in such a way that the discrepancy between the desired shape of the true posterior and that of the proposal is compensated. Accordingly, the importance weights in (2.9) are given as

$$w_{n-1}^{(i)} \propto \frac{p\left(\mathbf{x}_{n-1}^{(i)} | \mathbf{y}_{1:n-1}\right)}{\pi\left(\mathbf{x}_{n-1}^{(i)} | \mathbf{x}_{n-2}, \mathbf{y}_{1:n-1}\right)}, \tag{2.11}$$

where the symbol $\propto$ means "proportional to".

In light of the discrete summation in (2.9) and the importance weights in (2.11), let us discretize the predict/update equations given in (2.7) and (2.8) that are continuous in state. Then the prediction step in (2.7) becomes

$$p\left(\mathbf{x}_n | \mathbf{y}_{n-1}\right) \approx \sum_{i=1}^{M} p\left(\mathbf{x}_n | \mathbf{x}_{n-1}^{(i)}\right) w_{n-1}^{(i)} \delta\left(\mathbf{x}_{n-1} - \mathbf{x}_{n-1}^{(i)}\right). \tag{2.12}$$

To realize the update step in (2.8), let us update the time instant from $n-1$ to n in (2.11),

$$w_n^{(i)} \propto \frac{p\left(\mathbf{x}_n^{(i)} | \mathbf{y}_{1:n}\right)}{\pi\left(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_{1:n}\right)}. \tag{2.13}$$

Invoking the Bayes' rule given in (2.8) for the numerator of (2.13), the weight update equation (2.13) can be re-organized as,

$$\begin{aligned} w_n^{(i)} &\propto \frac{p\left(y_n|\mathbf{x}_n^{(i)}\right) p\left(\mathbf{x}_n^{(i)}|\mathbf{y}_{1:n-1}\right)}{\pi\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}, \mathbf{y}_{1:n}\right)} \\ &\propto w_{n-1}^{(i)} \frac{p\left(y_n|\mathbf{x}_n^{(i)}\right) p\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}\right)}{\pi\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}, \mathbf{y}_{1:n}\right)}. \end{aligned} \tag{2.14}$$

Consequently, the approximation to the true posterior at time n is obtained as

$$p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right) \approx \sum_{i=1}^{M} w_n^{(i)} \delta\left(\mathbf{x}_n - \mathbf{x}_n^{(i)}\right). \tag{2.15}$$

The selection of the proposal density is a major design aspect for a successful realization of a particle filter. An optimal importance function is shown to be available only for a restricted class of state-space models (Arulampalam *et al.*, 2002; Doucet *et al.*, 2000). A common practice in particle filters is to use the transition prior $p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right)$ as the proposal density such that $\pi\left(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{1:n}\right) = p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right)$ (Arulampalam *et al.*, 2002; Gordon *et al.*, 1993), which leads to simple implementation. With this selection of the importance function, the weight update equation in (2.14) is simplified to

$$w_n^{(i)} \propto w_{n-1}^{(i)} p\left(y_n|\mathbf{x}_n^{(i)}\right). \tag{2.16}$$

A challenge in the practical realization of the particle filters is that after a few iterations of (2.16), only a few particles have non-negligible weights (Arulampalam *et al.*, 2002), which is known as the *particle attrition* problem. A remedy to this problem is presented by drawing a new set of particles with uniform weights at each iteration. This idea is termed the *resampling*. The resampling operation can be performed in different ways by several resampling algorithms, such as the *systematic resampling*, *stratified resampling* or the *residual resampling* (see (Hol *et al.*, 2006) for an overview of several resampling techniques). We concern ourselves with *systematic resampling* (Arulampalam *et al.*, 2002; Gordon *et al.*, 1993) due to its simplicity and reduced computational cost. The underlying idea of systematic resampling is the following: At each particle position at time $n-1$, draw a new set of particles at time n, whose number is proportional to the weight of the particle at that position. This idea is graphically illustrated for 10 particles in Fig.2.4. Notice that the fourth particle from the left, $x_{n-1}^{(4)}$ has the largest weight at time $n-1$, and 3 new particles with uniform weights are drawn at its position for the next time step. The other particles are also replicated or removed proportionally to the weights of particles at respective positions at $n-1$. We note that without resampling, the weight of the fourth particle would get larger and larger at each iteration while the weights of the others would vanish. After the completion of the resampling operation, the particles are propagated to the next time step by using the proposal density, hence the particle diversity is introduced. Since we set the proposal density equal to the transition prior, the propagation step takes place according to the state equation in the first line of (2.6).

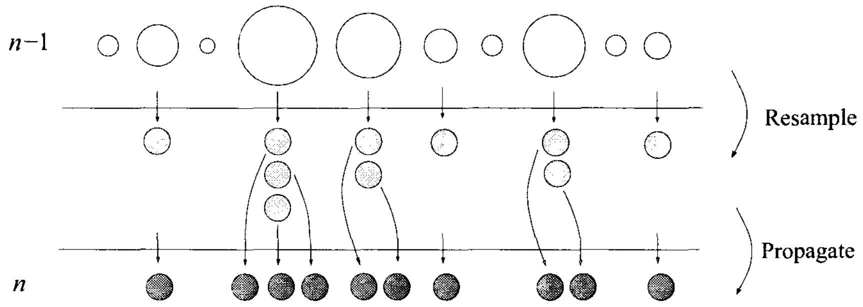


Figure 2.4: Graphical illustration of the systematic resampling of and propagation of particles from time $n-1$ to n.

Although the resampling and propagation operations help reduce the particle degeneration, the particles with large weights are still kept being replicated many times. This is called the *sample impoverishment*, which places certain limitations on the performances of the standard particle filters (SPF). Nevertheless, the SPF, which encompasses the selection of prior as the proposal, and the systematic resampling algorithm at each iteration, serves sufficiently well for our purposes. In the next subsection, we build a more advanced Bayesian technique called the Rao-Blackwellized particle filter (RBPF) on the SPF. The RBPF will be shown to provide superior performance for the applications that we are concerned with in Chapter4.

2.2.2 Rao-Blackwellized Particle Filter

In this subsection, we study the estimation of driving-forces that can be modeled as the additive components in the nonlinear state equations. To elaborate, imagine that the general description of a perturbed dynamic system in (1.1) is specified as

$$\begin{aligned} x_{n+1} =& g\left(x_n\right) + u_{n+1} + \omega_{n+1} \\ y_{n+1} =& h\left(x_{n+1}\right) + v_{n+1} \end{aligned} \tag{2.17}$$

where the driving-force u_{n+1} is an additive element in the governing body of the nonlinear state. Since any band-limited discrete time stochastic process can be described by an autoregressive (AR) model of arbitrary order (Papoulis, 1985), a linear submodel can be devised to describe the dynamic evolution of u_n such that,

$$u_{n+1} = c_1 u_n + \ldots + c_{\mathsf{P}} u_{n-\mathsf{P}+1} + \zeta_{n+1} \tag{2.18}$$

where ζ_n is white Gaussian noise (WGN) that accounts for the model input, whose variance is σ_ζ^2. The AR(P) model in (2.18) can be expressed with a Markovian transition in the vector form. For this transformation, let us collect the P driving-force samples from time n to $n - \mathsf{P} + 1$ in a vector and define this vector as,

$$\mathbf{u}_n \triangleq [u_n, \ldots, u_{n-\mathsf{P}+1}]^T \in \mathbb{R}^{\mathsf{P}}. \tag{2.19}$$

We refer to the vector $\mathbf{u}_n$ as the linear state vector. $\mathbf{u}_n$ can be regressed on its previous state by a $\mathsf{P}\times\mathsf{P}$ regression matrix, which holds the model coefficients $\{c_\mathsf{p}|\mathsf{p} = 1, \ldots, \mathsf{P}\}$ in the following form

$$\mathbf{B} = \begin{bmatrix} c_1 & c_2 & \cdots & c_{\mathsf{P}-1} & c_\mathsf{P} \\ 1 & 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{2.20}$$

This enables us to re-arrange (2.17) as given in (2.21), in which both the linear and the nonlinear state variables are represented in dynamic Markovian sub-structures,

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{B}\mathbf{u}_n + \xi_{n+1} \\ x_{n+1} &= g\left(x_n\right) + \mathbf{b}\mathbf{u}_n + \omega_{n+1} \\ y_{n+1} &= h\left(x_{n+1}\right) + v_{n+1}. \end{aligned} \tag{2.21}$$

The $(1 \times \mathsf{P})$ vector $\mathbf{b}$ in the second line of (2.21) is defined as $\mathbf{b} = [1, 0, \ldots, 0]$. The $(\mathsf{P} \times 1)$ noise vector is given by $\xi_n \triangleq [\zeta_n, 0 \ldots, 0]^T$. The concatenation of the linear state $\mathbf{u}_n$ and the nonlinear state x_n is equivalent to the standard description of the state vector in (2.6) such that $\mathbf{x}_n \equiv [\mathbf{u}_n^T, x_n]^T$.

Andrieu and Doucet (2000); Hendeby *et al.* (2007); Schön *et al.* (2005) demonstrate that the Rao-Blackwellized particle filter is a favorable technique for the estimation of linear state variables, whenever it is possible to partition the state equation into a linear and a nonlinear sub-model as in (2.21). The RBPF exploits the linear sub-structure by virtue of Rao-Blackwellization (or marginalization). In particular, the nonlinear state x_n is estimated with a particle filter, and the linear state $\mathbf{u}_n$ is then marginalized from the nonlinear state. This marginalization is operated by a Kalman filter (KF). Due to the linearity of the first and second lines of (2.21), and the Gaussianity of ξ_n and ω_n, the KF estimates of the linear states $\mathbf{u}_n$ are optimum. Therefore, the RBPF is partly optimum. This explains why the RBPF provides superior performance over the standard particle filters by Andrieu and Doucet (2000); Schön *et al.* (2005).

For a more formal description of the RBPF, let $p\left(\mathbf{u}_n, x_n | \mathbf{y}_{1:n}\right)$ denote the joint posterior of the linear and nonlinear state, which can be written in the form,

$$p\left(\mathbf{u}_n, x_n | \mathbf{y}_{1:n}\right) = p\left(\mathbf{u}_n | x_n\right) p\left(x_n | \mathbf{y}_{1:n}\right) \tag{2.22}$$

where we have made use of the fact that the measurements are independent of the linear state. Then the idea is to marginalize the posterior density of the linear state from the joint posterior in (2.22) as follows,

$$\begin{aligned} p\left(\mathbf{u}_n | \mathbf{y}_{1:n}\right) &= \int p\left(\mathbf{u}_n, x_n | \mathbf{y}_{1:n}\right) dx_n \\ &= \int p\left(\mathbf{u}_n | x_n\right) p\left(x_n | \mathbf{y}_{1:n}\right) dx_n. \end{aligned} \tag{2.23}$$

$p\left(x_n|\mathbf{y}_{1:n}\right)$ in the second line of (2.23) can be approximated by a particle filter. Specifically, let $\hat{p}\left(x_n|\mathbf{y}_{1:n}\right)$ denote the particle filter approximation to $p\left(x_n|\mathbf{y}_{1:n}\right)$, which can be described as given in (2.24),

$$\hat{p}\left(x_n|\mathbf{y}_{1:n}\right) = \sum_{i=1}^{M} w^{(i)}\delta\left(x_n - x_n^{(i)}\right). \tag{2.24}$$

Substituting (2.24) in the second line of (2.23), the approximate posterior density for the linear state is obtained as

$$\hat{p}\left(\mathbf{u}_n|\mathbf{y}_{1:n}\right) = \sum_{i=1}^{M} w^{(i)}p\left(\mathbf{u}_n|x_n^{(i)}\right). \tag{2.25}$$

The conditional mean estimates of the linear state variable are approximated from (2.25) in the following form:

$$\begin{aligned}\mathbb{E}\left[\mathbf{u}_n|\mathbf{y}_{1:n}\right] &\approx \int \mathbf{u}_n\hat{p}\left(\mathbf{u}_n|\mathbf{y}_{1:n}\right)d\mathbf{u}_n \\ &= \sum_{i=1}^{M} w^{(i)}\mathbb{E}\left[\mathbf{u}_n|x_n^{(i)}\right],\end{aligned} \tag{2.26}$$

where $\mathbb{E}\left[.\right]$ denotes the statistical expectation operation. $\mathbb{E}\left[\mathbf{u}_n|x_n^{(i)}\right]$ is computed using a Kalman filter exploiting the linear Gaussian substructure in the first and the second lines of (2.21).

The overall implementation of the Rao-Blackwellized particle filter is provided in a pseudo-code in Algorithm 1, where the nonlinear state is estimated with a standard particle filter. Hence, the importance function is set equal to the transition prior, and the systematic resampling is used at each iteration. The filter's computational complexity can be easily derived from the Algorithm 1 as will be explained in the

Appendix B.2.

Before explaining the steps of Algorithm 1, let us introduce the following notation for the estimated quantities, which will be commonly used in the sequel:

$\hat{\mathbf{u}}_{n|n}$: filtered estimates of the vector $\mathbf{u}_n$, and $\mathbf{C}_{n|n}$: filtering covariance for $\hat{\mathbf{u}}_{n|n}$,

$\hat{\mathbf{u}}_{n+1|n}$: one-step predicted estimates of the vector $\mathbf{u}_n$, and $\mathbf{C}_{n+1|n}$: prediction covariance for $\hat{\mathbf{u}}_{n+1|n}$,

$\hat{\mathbf{u}}_{n|\mathcal{T}}$: fixed-interval smoothed estimates of the vector $\mathbf{u}_n$ and $\mathbf{C}_{n+1|\mathcal{T}}$: smoothing covariance for $\hat{\mathbf{u}}_{n|\mathcal{T}}$. The fixed-interval smoothing implies that all the data available from time 1 to $\mathcal{T}$ are utilized to estimate the vector $\mathbf{u}_n$. The smoothed estimates are especially frequently used in Section 2.2.3.

Now let us have a closer look at the steps of the Algorithm 1. The first and the second tabs in the line 3 in Algorithm 1 are the Kalman prediction equations, where $\mathbf{R}$ stands for the $\mathsf{P} \times \mathsf{P}$ covariance matrix for the evolution uncertainty for the linear state vector, whose only nonzero entry is its first element, σ_ζ^2. For instance, letting $\mathsf{P} = 2$, we have

$$\mathbf{R} = \begin{bmatrix} \sigma_\zeta^2 & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.27}$$

The first tab of the line 4 denotes the sampling of particles from the transition prior, whereas the second tab of line 4 explains the derivation of the pseudo-measurements obtained from the nonlinear state equation.

The Kalman gain vector at time n is denoted Υ_n at line 6, which is a $\mathsf{P} \times 1$ column vector.

Line 7 describes the Kalman measurement update equations. The estimate of the linear state vector associated with the (i)th particle is denoted $\hat{\mathbf{u}}_{n|n}^{(i)}$ as given in the first tab of line 7. $\mathbf{C}_{n|n}$ in the second tab of line 7 denotes the $\mathsf{P} \times \mathsf{P}$ covariance matrix for the estimation error for $\hat{\mathbf{u}}_{n|n}^{(i)}$. Note that each particle is associated with a Kalman filter. However, the same covariance matrix is propagated for all Kalman filters, since the measurement equation is independent of the linear state vector. This greatly reduces the complexity of the RBPF.

Finally, the desired conditional mean estimates of the linear state vector are obtained by averaging over all particles as observed in line 8 since the importance weights have already been normalized. The filtered driving-force estimate, $\hat{u}_n$ equals the first element of $\hat{\mathbf{u}}_{n|n}$.

2.2.3 Expectation Maximization - Particle Filter (EM-PF)

The Rao-Blackwellized particle filter is intended for such applications where the exact knowledge of the system equations (e.g. (2.21)) is available. However, in some applications, a model of the underlying dynamic system may be only partially available (Andrieu and Doucet, 2000, 2003; Liu and West, 2001). In this subsection, we consider such cases where the nonlinear state equation and the measurement equation are available, yet the evolution of the linear state is hidden from the observer. We propose to use the *expectation maximization* (EM) (Dempster *et al.*, 1977) algorithm for such estimation scenarios, where the hidden states have to be inferred from incomplete data. The objective of the EM algorithm is to parameterize the missing information, and obtain maximum likelihood (ML) estimates of those parameters.

The driving-force estimator based on the EM algorithm is embodied by the following two main operations:

1. Particle Filtering: using the readily available system equations (i.e. the second

Algorithm 1 THE RAO-BLACKWELLIZED PARTICLE FILTER

```
Require: Observables from the environment $\{y_n | n = 1, \ldots, \mathcal{T}\}$ and an analytical description of the underlying mechanism (e.g. (2.21))
Ensure: Conditional mean estimates of the linear state vector; $\hat{\mathbf{u}}_{n|n}$ for $n = 1, \ldots, \mathcal{T}$
  Initialize particles, $i = 1, \ldots, M$; $x_{0|-1}^{(i)} \sim p(x_0)$ and set $\mathbf{u}_{0|-1}^{(i)} = \mathbf{u}_0^{(i)}$, and $\mathbf{C}_{0|-1}^{(i)} = \mathbf{C}_0$
  $for\ (n = 0; n < \mathcal{T}; t++)$ {
    $for\ (i = 1; i \leq M; i++)$ {
  Evaluate the importance weights $w_n^{(i)} = p\left(y_n | x_n^{(i)}\right)$ and normalize $\tilde{w}^{(i)} = w^{(i)} / \sum_{j=1}^{M} w_n^{(j)}$
  Resample $M$ particles with replacement, $\mathbb{P}\left(x_{n|n}^{(i)} = x_{n|n-1}^{(j)}\right) = \tilde{w}_n^{(j)}$
  State prediction and state prediction covariance:
      • $\hat{\mathbf{u}}_{n+1|n}^{(i)} = \mathbf{B}\hat{\mathbf{u}}_{n|n}^{(i)}$
      • $\mathbf{C}_{n+1|n} = \mathbf{B}\mathbf{C}_{n|n}\mathbf{B}^T + \mathbf{R}$
  For $i = 1, \ldots, M$,
      • New particle prediction $x_{n+1|n}^{(i)} \sim p\left(x_{n+1|n} | x_n^{(i)}\right) = \mathcal{N}\left(g\left(x_n^{(i)}\right) + \mathbf{b}\hat{\mathbf{u}}_{n|n}^{(i)}, \mathbf{b}\mathbf{C}_{n|n}\mathbf{b}^T + \sigma_\omega^2\right)$
      • Measurement prediction $m_{n+1}^{(i)} = x_{n+1}^{(i)} - g\left(x_n^{(i)}\right)$
  $\Omega_{n+1} = \mathbf{b}\mathbf{C}_{n+1|n}\mathbf{b}^T + \sigma_\omega^2$ {Innovation covariance}
  $\Upsilon_{n+1} = \mathbf{C}_{n+1|n}\mathbf{b}^T\Omega_{n+1}^{-1}$ {Kalman gain}
  Updated state estimate and updated state covariance:
      • $\hat{\mathbf{u}}_{n+1|n+1}^{(i)} = \hat{\mathbf{u}}_{n+1|n}^{(i)} + \Upsilon_{n+1}\left(m_{n+1}^{(i)} - \mathbf{b}\hat{\mathbf{u}}_{n+1|n}^{(i)}\right)$
      • $\mathbf{C}_{n+1|n+1} = \mathbf{C}_{n+1|n} - \Upsilon_{n+1}\Omega_{n+1}\Upsilon_{n+1}^T$
    }
  Desired estimates: $\hat{\mathbf{u}}_{n|n} = (1/M)\sum_{i=1}^{M} \hat{\mathbf{u}}_{n|n}^{(i)}$
  }
```

and the third lines of (2.21)), and the set of measurements $\mathbf{y}_{1:n}$, the nonlinear state x_n in (2.21) is estimated with a particle filter. The estimates of u_n provided by the EM algorithm are also substituted in (2.21).

2. Expectation Maximization: The EM algorithm operates on a sub– state-space model, which consists of
   - a state equation, (i.e. the first line of (2.21)), whose parameters are missing. This means that the first line of (2.21) is treated as an unknown equation by the EM, whose parameters are to be estimated;
   - a measurement equation, (i.e. the second line of (2.21)), in which the estimates of x_n provided by the particle filter are treated as the measurements.

Following from this brief description, we abbreviate the resulting method as the "EM-PF". The operation of the EM-PF method is illustrated on a block diagram as given in Fig.2.5. The EM-PF algorithm is proposed a doubly-iterative method that encompasses an inner and an outer loop of iterations. The inner loop consists of K EM iterations over the entire dataset. After the completion of one inner loop, the outer loop is re-initiated by running the particle filter all over again to exploit the updated estimates of u_n provided by the EM algorithm. Thus, the available data $\mathbf{m}$ are updated by the particle filter, and provided to the E-step of the EM algorithm.

For the particle filtering operations, we use the SPF, which is addressed in detail in Section 2.2.1. Now, let us elaborate on the EM method by starting with the construction of the sub– state-space pair given in (2.28), which is built on the first

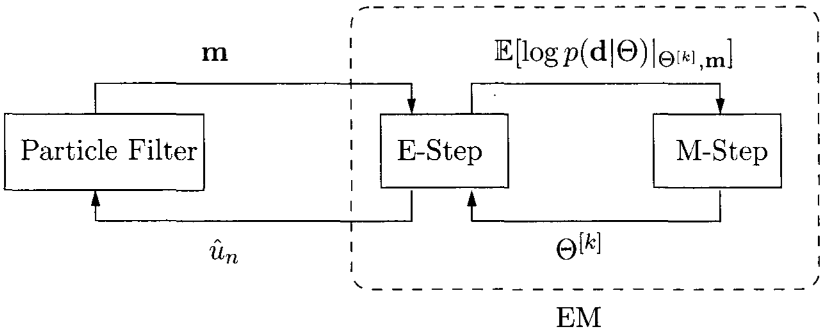

m
$\mathbb{E}[\log p(\mathbf{d}|\Theta)|_{\Theta^{[k]},\mathbf{m}}]$
Particle Filter
E-Step
M-Step
$\hat{u}_n$
$\Theta^{[k]}$
EM

Figure 2.5: Block diagram for the EM-PF method. The outer iteration loop includes the exchange of data between the particle filter and the E-step of the EM algorithm. The inner iteration loop represents the exchange of data between the E and the M-steps of the EM algorithm.

and the second lines of (2.21):

$$\begin{aligned} \mathbf{u}_n =& \mathbf{B}\mathbf{u}_{n-1} + \xi_n \\ m_n =& \mathbf{b}\mathbf{u}_n + \omega_n. \end{aligned} \tag{2.28}$$

In (2.28), we have defined $m_n \triangleq x_n - g(x_{n-1})$, which are treated as the pseudo-measurements for the EM algorithm. Note that the model in (2.28) is suitable for the operation of a Kalman smoother (KS), which is needed to perform the expectation (E) step of the EM algorithm as will be explained soon. In the M-step, the log-likelihood of the complete data is maximized with respect to the unknown coefficients. Since the unknown regression coefficients $\{c_{\mathsf{p}}|\mathsf{p} = 1, \ldots, \mathsf{P}\}$ occupy only the first row of the matrix $\mathbf{B}$, performing the M-step over the model in (2.28) would be wasteful of computational resources. As an alternative, we simplify (2.28) to obtain (2.29), where we deal with the $1 \times \mathsf{P}$ vector of unknown coefficients ($\mathbf{c} \triangleq [c_1, \ldots, c_{\mathsf{P}}]$) instead of the

$\mathsf{P} \times \mathsf{P}$ matrix, $\mathbf{B}$:

$$\begin{aligned} u_n =& \mathbf{c}\mathbf{u}_{n-1} + \zeta_n \\ m_n =& u_n + \omega_n. \end{aligned} \tag{2.29}$$

Consequently, the E-step of the EM algorithm operates on (2.28), whereas the M-step exploits (2.29).

Imagine that the conditional mean estimates of the nonlinear state x_n are obtained with a particle filter from time $n = 1$ to $n = \mathcal{T}$. For the sake of brevity, let us assume that the particle filter estimates are arbitrarily close to the true states. Then let us collect the pseudo-measurements in a $\mathcal{T} \times 1$ vector such that $\mathbf{m} \triangleq [m_1, \ldots, m_{\mathcal{T}}]^T$, which represents the available data. Let us also concatenate the entire history of the driving-force samples in another $\mathcal{T} \times 1$ vector, $\mathbf{U} \triangleq [u_1, \ldots, u_{\mathcal{T}}]^T$. The vector $\mathbf{U}$ represents the missing data which are associated with the hidden parameters, $\Theta \triangleq [c_1, \ldots, c_{\mathsf{P}}, \sigma_\zeta^2]^T$. The complete data are represented as a concatenation of the missing and the available data. Let the vector $\mathbf{d}$ denote the complete data, which is defined as

$$\mathbf{d} \triangleq [\mathbf{U}^T, \mathbf{m}^T]^T. \tag{2.30}$$

The E and the M-steps of the EM algorithm are iterated until convergence. In the E-step, the expectation of the log-likelihood of complete data is simulated given the available data $\mathbf{m}$, and the current estimates of the hidden parameters, $\Theta^{[k]}$. In the M-step, the expected log-likelihood of the complete data is maximized with respect to the hidden parameter vector. This maximization results in an updated ML estimate of Θ.[1]

Let us describe the log-likelihood of the complete data as

$$\begin{aligned}\log p\left(\mathbf{d}|\Theta\right) &= \log p\left(\mathbf{U},\mathbf{m}|\Theta\right) \\ &= \log p\left(\mathbf{U}|\Theta\right) + \log p\left(\mathbf{m}|\mathbf{U},\Theta\right).\end{aligned} \tag{2.31}$$

Exploiting the linearity of our sub-model in (2.29) and using the whiteness and Gaussianity of ζ_n and ω_n, the second line of (2.31) can be expressed as

$$\begin{aligned}\log p\left(\mathbf{d}|\Theta\right) =& \log p\left(u_0\right) - \frac{\mathcal{T}}{2}\log\sigma_\zeta^2 - \mathcal{T}\log 2\pi - \frac{\mathcal{T}}{2}\log\sigma_\omega^2 - \\ & \frac{1}{2\sigma_\zeta^2}\sum_{n=1}^{\mathcal{T}}\left(u_n - \mathbf{c}^T\mathbf{u}_{n-1}\right)^2 - \frac{1}{2\sigma_\omega^2}\sum_{n=1}^{\mathcal{T}}\left(m_n - u_n\right)^2.\end{aligned} \tag{2.32}$$

Using (2.32), we are ready to elaborate on the E and the M-steps:

- E-step: Let $\Theta^{[k]}$ be an estimate of the parameters at the kth EM iteration. Then for the E-step, we compute the expectation of the log-likelihood in (2.32)

[1] Another popular interpretation of the EM algorithm is the maximization of the lower bound on the log-likelihood of the incomplete data (Roweis and Ghahramani, 2001). This interpretation is more adequate for nonlinear, non-Gaussian models, in which case the conditional density of the missing data needs to be simulated for the E-step. Nevertheless, the M-step in the two interpretations in (Dempster *et al.*, 1977) and (Roweis and Ghahramani, 2001) can be shown to be identical when the variational distribution of the missing data in (Roweis and Ghahramani, 2001) is set to the actual conditional density of the missing data, in which case the lower bound is satisfied with equality.

given the available data $\mathbf{m}$, and our current parameter estimate, $\Theta^{[k]}$:

$$\begin{aligned}
E\left(\Theta|\Theta^{[k]}\right) \triangleq & \mathbb{E}\left[\log p(\mathbf{d}|\Theta)|_{\mathbf{m},\Theta^{[k]}}\right] \\
= & o_1 - \frac{\mathcal{T}}{2}\log\sigma_\zeta^2 \\
& -\frac{1}{2\sigma_\zeta^2}\sum_{n=1}^{\mathcal{T}}\mathbb{E}\left[(u_n)^2|_{\mathbf{m},\Theta^{[k]}}\right] - 2\mathbf{c}^T\mathbb{E}\left[u_n\mathbf{u}_{n-1}|_{\mathbf{m},\Theta^{[k]}}\right] + \mathbf{c}^T\mathbb{E}\left[\mathbf{u}_{n-1}\mathbf{u}_{n-1}^T|_{\mathbf{m},\Theta^{[k]}}\right]\mathbf{c} \\
& -\frac{1}{2\sigma_\omega^2}\sum_{n=1}^{\mathcal{T}}\mathbb{E}\left[(m_n - u_n)^2|_{\mathbf{m},\Theta^{[k]}}\right]
\end{aligned} \tag{2.33}$$

where we have defined the constant term as $o_1 \triangleq \log p(u_0) - \mathcal{T}\log 2\pi - \frac{\mathcal{T}}{2}\log\sigma_\omega^2$. As noted before, the E-step operates on the model in (2.28), and the indicated expectations in (2.32) are approximated using a KS. For instance, in order to compute the term $\mathbb{E}\left[(u_n)^2|_{\mathbf{m},\Theta^{[k]}}\right]$, we first compute $\mathbb{E}[\mathbf{u}_n\mathbf{u}_n^T|_{\mathbf{m},\Theta^{[k]}}]$ using a KS. Then $\mathbb{E}\left[(u_n)^2|_{\mathbf{m},\Theta^{[k]}}\right]$ is retrieved from the first element of $\mathbb{E}[\mathbf{u}_n\mathbf{u}_n^T|_{\mathbf{m},\Theta^{[k]}}]$. The operation of the KS is described in full detail in line 3 of the Algorithm 2.

- M-step: We find the value of Θ that maximizes $E\left(\Theta|\Theta^{[k]}\right)$, which becomes the next estimate of the missing parameters,

$$\Theta^{[k+1]} = \arg\max_{\Theta} E\left(\Theta|\Theta^{[k]}\right) \tag{2.34}$$

where $E\left(\Theta|\Theta^{[k]}\right)$ is as given in (2.33). The details and the outcome of the maximization operation are provided in the line 2 of the Algorithm 2.

Note that the KS in the E-step is used for approximating the second-order quantities indicated in the expectations in (2.33). As a side product of the E-step, the smoothed driving-force estimates can also be retrieved from the KS, which is the case in our application.

It is shown in (Roweis and Ghahramani, 2001) that the likelihood of the missing parameters cannot decrease during the EM iterations. Hence, at each iteration, the expectations in the E-step are performed over more accurate estimates of the model parameters, Θ. This implies that the quality of the driving-force estimates are either improved or not decreased at each EM iteration.

The performance of the particle filter has a crucial impact on the overall performance of the EM algorithm. The EM-PF algorithm will be put to test in Chapter 4 as well as the other driving-force estimation methods.

The entire realization of the EM-PF is given in a pseudo-code in Algorithm 2. The total number of EM iterations in Algorithm 2 is denoted by K. Although there does not exist an analytical way of finding the optimum value of K, in practice, we have observed that more than $4-6$ EM iterations do not lead to a significant improvement in the estimates of u_n. These results match with the observations of other researchers in the literature (Zia *et al.*, 2007, 2008).

Algorithm 2 THE EXPECTATION MAXIMIZATION - PARTICLE FILTER

```
Require: Observables from the environment $\{y_n | n = 1, \ldots, \mathcal{T}\}$ and an analytical
    description of the system (e.g. (2.28))
Ensure: Estimates of the missing parameters $\Theta$ and the smoothed conditional mean
    estimates of the driving-force; $\hat{u}_n$ for $n = 1, \ldots, \mathcal{T}$
    Initialize particles, $i = 1, \ldots, M$; $x^{(i)}_{0|-1} \sim p(x_0)$ and set $\mathbf{u}^{(i)}_{0|-1} = \mathbf{u}^{(i)}_0$, and $\mathbf{C}^{(i)}_{0|-1} = \mathbf{C}_0$
    Initialize the estimate for the missing parameters: $\Theta^{[0]}$
    $for\ (k = 0; k < K; k++)\ \{$
        $for\ (n = 0; n < \mathcal{T}; t++)\ \{$
            $for\ (i = 1; i \leq M; i++)\ \{$
    Evaluate the importance weights $w^{(i)}_n = p\left(y_n | x^{(i)}_n\right)$ and normalize $\tilde{w}^{(i)} = w^{(i)} / \sum_{j=1}^{M} w^{(j)}_n$
    Res. $M$ particles with replacement, $\mathbb{P}\left(x^{(i)}_{n|n} = x^{(j)}_{n|n-1}\right) = \tilde{w}^{(j)}_n$
    E-STEP (Kalman smoother):
        1. Forward Recursions:
           State prediction and state prediction covariance:
            • $\hat{\mathbf{u}}^{(i)}_{n+1|n} = \mathbf{B}^{[k]} \hat{\mathbf{u}}^{(i)}_{n|n}$
            • $\mathbf{C}_{n+1|n} = \mathbf{B}^{[k]} \mathbf{C}_{n|n} \left(\mathbf{B}^{[k]}\right)^T + \mathbf{R}^{[k]}$
            • Particle prediction $x^{(i)}_{n+1|n} \sim p\left(x_{n+1|n} | x^{(i)}_n\right) = \mathcal{N}\left(g\left(x^{(i)}_n\right) + \mathbf{b}\hat{\mathbf{u}}^{(i)}_{n|n}, \mathbf{b}\mathbf{C}_{n|n}\mathbf{b}^T + \sigma^2_\omega\right)$
            • Measurement prediction $m^{(i)}_{n+1} = x^{(i)}_{n+1} - g\left(x^{(i)}_n\right)$
            • $\zeta_{n+1} = \mathbf{b}\mathbf{C}_{n+1|n}\mathbf{b}^T + \sigma^2_\omega$ {Innovation covariance}
            • $\Omega_{n+1} = \mathbf{C}_{n+1|n}\mathbf{b}^T \zeta^{-1}_{n+1}$ {Kalman gain}
              Updated state estimate and state covariance:
            • $\hat{\mathbf{u}}^{(i)}_{n+1|n+1} = \hat{\mathbf{u}}^{(i)}_{n+1|n} + \Omega_{n+1}\left(m^{(i)}_{n+1} - \mathbf{b}\hat{\mathbf{u}}^{(i)}_{n+1|n}\right)$
            • $\mathbf{C}_{n+1|n+1} = \mathbf{C}_{n+1|n} - \Omega_{n+1}\zeta_{n+1}\Omega^T_{n+1}$
                        } }
Algorithm 2 Cont'd
  2. Backward Recursions:
     $for\ (n = \mathcal{T}; t \geq 0; t--)\ \{$
        $for\ (i = 1; i \leq M; i++)\ \{$

           • $\mathbf{P}_n = \mathbf{C}_{n|n} \left(\mathbf{B}^{[k]}\right)^T \mathbf{C}_{n+1|n}^{-1}$
           • $\hat{\mathbf{u}}_{n|\mathcal{T}}^{(i)} = \hat{\mathbf{u}}_{n|n}^{(i)} + \mathbf{P}_n \left(\hat{\mathbf{u}}_{n+1|\mathcal{T}}^{(i)} - \hat{\mathbf{u}}_{n+1|n}^{(i)}\right)$
           • $\mathbf{C}_{n|\mathcal{T}} = \mathbf{C}_{n|n} + \mathbf{P}_n \left(\mathbf{C}_{n+1|\mathcal{T}} - \mathbf{C}_{n+1|n}\right)$
                                        }
           • $\hat{\mathbf{u}}_{n|\mathcal{T}} = (1/M) \sum_{i=1}^{M} \hat{\mathbf{u}}_{n|\mathcal{T}}^{(i)}$ → Desired estimates: $\mathbb{E}[\hat{\mathbf{u}}_{n|\mathcal{T}}]$
           • $\mathbb{E}\left[\hat{\mathbf{u}}_n \left(\hat{\mathbf{u}}_n\right)^T\right] = \mathbf{C}_{n|\mathcal{T}} + \hat{\mathbf{u}}_{n|\mathcal{T}} \left(\hat{\mathbf{u}}_{n|\mathcal{T}}\right)^T$
           • $\mathbb{E}\left[\hat{\mathbf{u}}_{n-1} \left(\hat{\mathbf{u}}_{n-1}\right)^T\right] = \mathbf{C}_{n-1|\mathcal{T}} + \hat{\mathbf{u}}_{n-1|\mathcal{T}} \left(\hat{\mathbf{u}}_{n-1|\mathcal{T}}\right)^T$
           • $\mathbb{E}\left[\hat{\mathbf{u}}_n \left(\hat{\mathbf{u}}_{n-1}\right)^T\right] = \mathbf{P}_{n-1}\mathbf{C}_{n|\mathcal{T}} + \hat{\mathbf{u}}_{n|\mathcal{T}} \left(\hat{\mathbf{u}}_{n-1|\mathcal{T}}\right)^T$
                                     }
  4: M-STEP:
     • $\left(\mathbf{c}^{[k+1]}\right)^T =$
       $\sum_{n=1}^{\mathcal{T}} \mathbb{E}\left[u_n \left(\hat{\mathbf{u}}_{n-1}\right)^T |_{\mathbf{m},\Theta^{[k]}}\right] \left(\sum_{n=1}^{\mathcal{T}} \mathbb{E}\left[\hat{\mathbf{u}}_{n-1} \left(\hat{\mathbf{u}}_{n-1}\right)^T |_{\mathbf{m},\Theta^{[k]}}\right]\right)^{-1}$
     • $\sigma_{\xi}^{2,[k+1]} = \frac{1}{\mathcal{T}} \sum_{n=1}^{\mathcal{T}} \left(\mathbb{E}\left[(u_n)^2\right] - 2\left(\mathbf{c}^{[k]}\right)^T \left(\mathbb{E}\left[u_n \left(\hat{\mathbf{u}}_{n-1}\right)^T\right]\right)^T +\right.$
       $\left.\left(\mathbf{c}^{[k]}\right)^T \mathbb{E}\left[\hat{\mathbf{u}}_{n-1} \left(\hat{\mathbf{u}}_{n-1}\right)^T\right] \mathbf{c}^{[k]}|_{\mathbf{m},\Theta^{[k]}}\right)$
     • Allocate $\left(\mathbf{c}^{[k+1]}\right)^T$ into the first row of $\mathbf{B}^{[k+1]}$ and $\sigma_{\xi}^{2,[k+1]}$ into the first element
       of $\mathbf{R}^{[k+1]}$
                          }
```

Chapter 3

The Adaptive Driving-Force Estimator

In this chapter, the methodology for the proposed driving-force estimator, demonstrations of the algorithm's performance on some controlled experiments, and some applications of the proposed method to the real-world problems are presented. Most of the material in this chapter is a reproduction of the contents of the two journal articles, (Güntürkün, 2010a), (Güntürkün, 2010b). Section 3.1 and Section 3.2 present the material in the two articles respectively in a coherent manner. The overlapping parts of the two articles have been removed in the following presentation.

3.1 Sequential Reconstruction of Driving-Forces From Nonlinear Nonstationary Dynamics

3.1.1 Introduction

This section describes the proposed adaptive driving-force estimator (Güntürkün, 2010a). The driving-forces account for the perturbation inputs induced by the external environment or the secular variations in the internal variables of the system. The proposed algorithm is applicable to the problems for which there is too little or no prior knowledge to build a rigorous mathematical model of the unknown dynamics. We derive the estimator conditioned on the differentiability of the unknown system's mapping, and smoothness of the driving-force. The proposed algorithm is an adaptive sequential realization of the blind prediction error method, where the basic idea is to predict the observables, and retrieve the driving-force from the prediction error. Our realization of this idea is embodied by predicting the observables one-step into the future using a bank of echo state networks (ESN) in an online fashion, and then extracting the raw estimates from the prediction error and smoothing these estimates in two adaptive filtering stages. The adaptive nature of the algorithm enables the retrieval of both slowly and rapidly varying driving-forces accurately, which are illustrated by simulations. Logistic and Moran-Ricker maps are studied in controlled experiments, exemplifying chaotic state and stochastic measurement models. The algorithm is also applied to the estimation of a driving-force from another nonlinear dynamic system that is stochastic in both state and measurement equations. The results are judged by the posterior Cramer-Rao lower bounds. The method is finally put into test on a real-world application; extracting sun's magnetic flux from the sunspot time series.

In Section 3.1.2, we address the problem in a formal way, and present the derivation of the reconstruction algorithm. We reproduce the generalized posterior Cramer-Rao lower bound (PCRB) in Section 3.1.3. In Section 3.1.4, we present the results of controlled experiments, which are evaluated using the PCRB. In Section 3.1.5, we apply the estimator to the real-life sunspot time series for the reconstruction of sun's magnetic flux. We compare our results to two other models, and show that the proposed approach captures the essential dynamics of the sun's magnetic flux. Section 3.1.6 concludes the section with remarks on the future research.

In Section 3.2, we propose a solution to the radar scene analysis problem using the proposed ADFE and exploiting the texture modeling of sea clutter. The radar scene analysis problem is addressed in Section 3.2.1 for the design of a cognitive radar receiver. The texture modeling of the sea clutter is briefly reproduced in Section 3.2.2. In Section 3.2.3, a tracking example is studied. Section 3.2.4 is devoted to the description of the Dartmouth database, and the results of the experiments conducted on live recorded data. Section 3.3 concludes the section.

3.1.2 Regularized Estimation and Tracking of Unobservable Inputs

The basic idea underlying our approach is to predict the observables one-step into the future using a bank of echo state networks (ESN) in an online fashion, and then exploit the fact that the driving-force is not explicitly modeled by the predictor. Based on this simple idea, we relate the driving-force to the prediction error of the ESNs.

A predictive model of the unknown environment can be built by transforming the evolution of the observables y_n from the unknown system in (1.1) into a time series model. In such a model, the requirement for knowledge of the equations in (1.1), and the dependence of the observables on the unknown state is eliminated. For derivation, let us begin with re-arranging the second line of (1.1),

$$x_n = h^{-1}\left(y_n - v_n\right). \tag{3.1}$$

Now let us substitute (3.1) in the state equation in the first line of (1.1):

$$x_{n+1} = g\left(h^{-1}\left(y_n - v_n\right), u_n\right) + \omega_{n+1}. \tag{3.2}$$

If we plug (3.2) in the measurement equation in the second line of (1.1), we obtain the observable as

$$y_{n+1} = h\left(g\left(h^{-1}\left(y_n - v_n\right), u_n\right) + \omega_{n+1}\right) + v_{n+1}. \tag{3.3}$$

Let us denote the nested combination of mappings by $f(\alpha, \beta) \triangleq h\left(g\left(h^{-1}(\alpha), \beta\right)\right)$ with the aid of two dummy variables, α and β. Then, we can write

$$y_{n+1} \approx f\left(y_n, u_n\right) + \upsilon_{n+1}, \tag{3.4}$$

where the overall effect of the transformed and the additive noise processes is denoted by υ_{n+1}. In (3.4), the observable y_{n+1} is expressed as a function of the previous observable y_n, and the unknown driving-force, u_n.

Next, consider that we design an online nonlinear predictor (i.e. an ESN bank as described in Section 2.1), which receives only the previous observable in its input and provides one-step approximations to the unknown system. Let z_{n+1} denote the output of the nonlinear predictor, which is described as

$$z_{n+1} = \widetilde{f}\left(y_n, 0\right), \tag{3.5}$$

where y_n is input to the predictor, and z_{n+1} is obtained at its output. Note that the notation $\widetilde{f}\left(y_n, 0\right)$ implies that the driving-force is not provided to the neural network. Therefore, the absence of $u_n$at the neural network input is denoted by 0 as a second argument. The predictor's trainable parameters are updated at each time instant based on the arrival of a new observable. Then the objective is to extract the driving-force signal from the online prediction error $e_{n+1} = y_{n+1} - z_{n+1}$ based on the finite-difference approximation, given that the mapping f is differentiable everywhere:

$$\begin{aligned} e_{n+1} &= f\left(y_n, u_n\right) - \widetilde{f}\left(y_n, 0\right) + v_{n+1} \\ &\approx \frac{\partial f\left(y_n, u_n\right)}{\partial u_n} u_n + \epsilon_{n+1}. \end{aligned} \tag{3.6}$$

In (3.6), ϵ_{n+1} denotes the overall approximation error. The derivation of (3.6) will be elaborated in the sequel on the basis of some functional analysis theorems. The final step of the algorithm is to obtain the refined driving-force estimates using a regularized adaptive filter based on (3.6).

To sum up, the idea is to exploit the fact that the driving-force is not explicitly modeled by the predictor as suggested by Szeliga *et al.* (2002, 2003a,b); Verdes *et al.* (2001, 2004, 2006). Based on this fact, we relate the 1-step prediction error to the hidden input. The resulting estimation algorithm encompasses the following three main operations:

1. Predictive modeling of the unknown system using ESNs,
2. Linearization of the 1-step prediction error as in (3.6),
3. Noise smoothing and regularized adaptive estimation of the hidden input from the linearized prediction error.

One iteration of the algorithm is depicted in the block diagram in Fig.3.1.

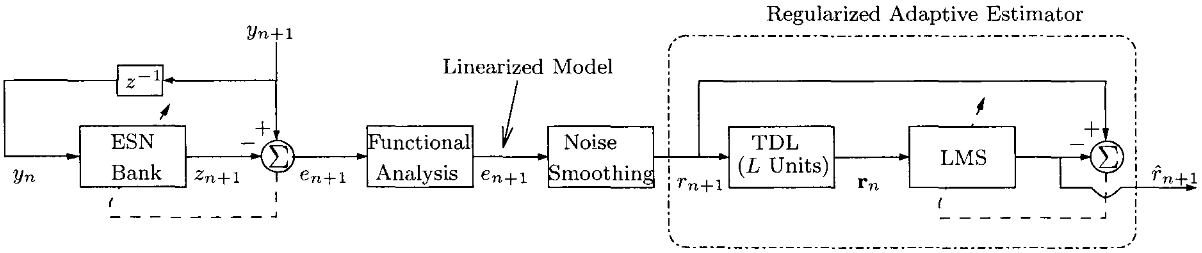


Figure 3.1: Block diagram of the ADFE. The task of the functional analysis block is to provide a model in which the prediction error e_{n+1} is linearly dependent on the driving-force. z^{-1} denotes the unit delay operator. "TDL" denotes a tapped-delay line of L shift registers. "LMS" stands for the least mean squares algorithm. $\mathbf{r}_n \triangleq [r_n, \dots, r_{n-L+1}]^T$.

In the remainder of this section, we elaborate on the three operations that form the basis of the algorithm.

Predictive Modeling of The Unknown System

A detailed treatment of the ESN has been provided in Section 2.1. Herein, we illustrate some numerical results to demonstrate the online prediction capability of the ESN bank for the applications studied in this chapter.

As stated in Section 2.1, the 1-step mean square prediction error averaged over all ESNs is found to be fairly insensitive to the selection of the free parameters of ESN.

To demonstrate this fact, we calculate the time-averaged mean squared error (MSE) on four different datasets that are studied in detail in Section 3.1.4 and 3.1.5. We measure the overall 1-step prediction error averaging over the individual prediction errors of each ESN in the bank. Let $e_{n+1}^{(i)} = y_{n+1} - z_{n+1}^{(i)}$ denote the 1-step prediction error of the ith ESN in the bank, $i = 1, 2, \dots, 100$. Our motivation in using a large number of ESNs (i.e. $V = 100$) herein is to demonstrate the algorithm's performance when arbitrarily large computational resources are available. In Chapter 4 however, we place some constraints on the computational cost, and analyze the performance of the driving-force estimator when the algorithm has to be realized with a much smaller amount of computaional resources.

The MSE in Fig.3.2 is estimated over 100 independent trials for each parameter value. Sensitivity of the algorithm to the selection of the ESN free parameters is analyzed in Fig.3.2. Inspecting Fig.3.2, it can be argued that the connectivity rate (ς) and the number of neurons (N) in DR do not play a crucial role on the performance of the network. The spectral radius (ρ) of the internal weight matrix however has a more remarkable effect. Nevertheless, the effect of ρ is not large enough to impede the prediction performance of the network significantly. As a result, we shall fix all the ESN parameters for the experiments in Section 3.1.4 and 3.1.5 without employing any search to find an optimum configuration.

The universal approximation theorem for recurrent neural networks (Jin *et al.*, 1995; Madan *et al.*, 1995) states that $f(y_n, u_n) : \mathbb{R}^2 \rightarrow \mathbb{R}$ in (3.4), whose initial values belong to a compact set D, can be approximated arbitrarily well by recurrent neural networks of finite order.

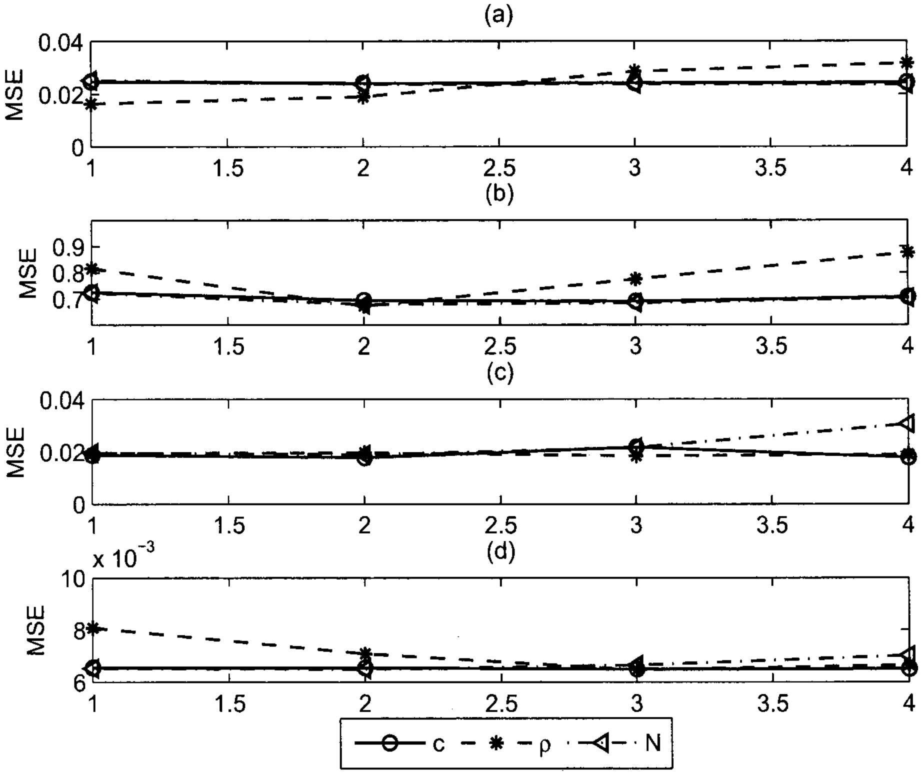


Figure 3.2: Time-averaged 1-step prediction MSE vs. free parameters of ESN. (a) Logistic map dataset. (b) Moran-Ricker dataset. (c) Nonlinear stochastic system. (d) Sunspot time series. The lines marked with circle, star and the triangle refer to the parameters ρ, ς, and N respectively. The x-axis is in arbitrary units and refers to the index of the free parameter under consideration. The following candidate values are tested: $\rho = [0.1, 0.3, 0.7, 0.9]$; $\varsigma = [4\%, 6\%, 8\%, 10\%]$; $N = [30, 40, 100, 200]$.

Linearized Approximate Model

Let us reproduce the 1-step prediction error in the first line of (3.6),

$$\begin{aligned} e_{n+1} &= y_{n+1} - z_{n+1} \\ &= f(y_n, u_n) - \tilde{f}(y_n, 0) + v_{n+1}. \end{aligned} \tag{3.7}$$

where $f(y_n, u_n)$ and $\tilde{f}(y_n, 0)$ are described in (3.4) and (3.5) respectively. In order to proceed further ahead with (3.7), we present Proposition 1:

Proposition 1. *Let $f \in B$ be differentiable everywhere. Hence, f is Lipschitz continuous. Let $\tilde{f} \in A$ be a mapping in a dense subset $A \subset B$. Then the following holds:*

$$\left| f(y_n, 0) - \tilde{f}(y_n, 0) \right| \leq \delta, \tag{3.8}$$

where $f(y_n, 0)$ represents the unperturbed (stationary) dynamics, and $\tilde{f}(y_n, 0)$denotes the neural network's modeling without the availability of the driving-force input.

The important implication of Proposition 1 is the following: Based on the Lipschitz continuity of f; the neural network response $\tilde{f}(y_n, 0)$ lies in a finite distance from the stationary dynamics mapped by $f(y_n, 0)$. Then formally, we can write $f(y_n, 0) = \tilde{f}(y_n, 0) + \varepsilon_{n+1}$, where ε_{n+1} is upper bounded such that $|\varepsilon_{n+1}| \leq \delta$ (which also follows from Proposition 1).

Proof. Let us subtract and add $f(y_n, u_n)$ on the left-hand side of (3.8) to obtain:

$$\left| f(y_n, 0) - \tilde{f}(y_n, 0) \right| = \left| f(y_n, 0) - f(y_n, u_n) + f(y_n, u_n) - \tilde{f}(y_n, 0) \right|. \tag{3.9}$$

Applying the triangle inequality to the right-hand side of (3.9), we get

$$\left| f(y_n, 0) - \tilde{f}(y_n, 0) \right| \leq \left| f(y_n, 0) - f(y_n, u_n) \right| + \left| f(y_n, u_n) - \tilde{f}(y_n, 0) \right|. \tag{3.10}$$

Due to the Lipschitz continuity of f, we can state $|f(y_n, 0) - f(y_n, u_n)| \leq \mathcal{L} |u_n|$ for the first term on the right-hand side of (3.10), which requires that the driving-force is upper-bounded. $\mathcal{L}$ is the Lipschitz constant. Also, exploiting the density of the neural network class (Zeidler, 1995, ch.1), we can state $\left|f(y_n, u_n) - \widetilde{f}(y_n, 0)\right| \leq \delta_1$ for the second term in right-hand side of (3.10). Then (3.10) becomes:

$$\left|f(y_n, 0) - \widetilde{f}(y_n, 0)\right| \leq \mathcal{L}|u_n| + \delta_1 \tag{3.11}$$

Setting $\delta = \mathcal{L}|u_n| + \delta_1$ completes the proof. □

Now, let us plug $\widetilde{f}(y_n, 0) = f(y_n, 0) - \varepsilon_n$ in the second line of (3.7) to obtain:

$$e_{n+1} = f(y_n, u_n) - f(y_n, 0) + \varepsilon_{n+1} + v_{n+1}. \tag{3.12}$$

Using the differentiability of f, let us replace the difference $[f(y_n, u_n) - f(y_n, 0)]$ in (3.12) by the partial derivative expression $\left[\frac{\partial f(y_n, u_n)}{\partial u_n}\right] u_n$ using the finite difference-approximation. This approximation will also introduce some errors. Then denoting the combination of all errors by ϵ, we have

$$e_{n+1} = \frac{\partial f(y_n, u_n)}{\partial u_n} u_n + \epsilon_{n+1}. \tag{3.13}$$

Now that we have a linearized system from which the hidden input is to be reconstructed.

Estimation of Hidden Input Using Wold's Decomposition

As a first step toward the derivation of the estimator, we reduce the excess noise in (3.13) using an exponentially decaying noise smoother. At this point, we stress the notation on the four key estimation quantities to clarify the notation in the sequel:

- u_n: the driving-force,

- e_n: the prediction error obtained by the ESN bank,
- r_n: the output of the noise-smoothing filter,
- $\hat{r}_n$: the output of the regularized adaptive estimator.

The input-output relation for the noise-smoothing filter is given by

$$r_{n+1} = (1 - \mathit{\Gamma})e_{n+1} + \mathit{\Gamma} r_n \tag{3.14}$$

where e_{n+1} is the filter input, and the filter output is denoted by r_{n+1}. The objective of this filter is to reduce the variance of the additive noise, ϵ_n in (3.13) by suppressing the high frequency noise components. The output of the noise-smoother is further refined using a regularized adaptive estimator, which is described in the sequel. Note that the task of the noise smoother is to cancel very high frequency components, which are assumed to be noise contributions. On the other hand, the signal component is assumed to occupy a lower bandwidth. Therefore, although the selection of $\mathit{\Gamma}$ can be fine-tuned for specific applications, we do not search for an optimum value for $\mathit{\Gamma}$, and set $\mathit{\Gamma} = 0.9$ throughout the experiments as a rule of thumb. Let us describe the relation between the output of the noise-smoothing filter and the driving-force as given in (3.15)

$$r_{n+1} = \frac{\partial f\left(y_n, u_n\right)}{\partial u_n} u_n + \epsilon'_{n+1}, \tag{3.15}$$

where ϵ' stands for the reduced noise. We derive the adaptive estimator based on Wold's decomposition theorem, which states the following (Papoulis, 1985):

Theorem 1. *Wold's Decomposition*

Any arbitrary discrete-time stochastic process can be decomposed into the summation of an unpredictable and a predictable part such that $\frac{\partial f(y_n, u_n)}{\partial u_n} = o_n + \eta_n$, *where* o_n *is the stochastic part and* η_n *is predictable.*

Then (3.15) can be rearranged as $r_{n+1} = (o_n + \eta_n) u_n + \epsilon'_{n+1}$ deducing from Theorem 1. Denoting the stochastic part $\vartheta_{n+1} \triangleq \left[o_n u_n + \epsilon'_{n+1}\right]$, we obtain

$$r_{n+1} = \eta_n u_n + \vartheta_{n+1}. \tag{3.16}$$

Papoulis (1985) showed that there exists a predictor of r_{n+1} of the form $\hat{r}_{n+1} = \sum_{l=1}^{\infty} a_l r_{n-l+1}$, which is also a (non-unique) a predictor of $\eta_n u_n$ in (3.16). Estimating u_n from r_{n+1} is however an inverse problem, which is most likely to be ill-posed due to the abrupt variations in η_n, and the effect of convolutional noise, ϑ_n. Therefore, to obtain a practically meaningful estimator, we need to:

- bring some form of regularization into the solution of problem, and
- employ an adaptive mechanism so as to respond to the changes in the environmental conditions represented by the variations in η_n.

The method that we follow in this chapter is Tikhonov regularization (Tikhonov and Arsenin, 1977) which imposes a smoothness constraint on the driving-force. The driving-force is estimated using a regularized adaptive filter. Let us introduce the following notations for the filter parameters at discrete time n:

$\mathbf{r}_n = \left[r_n, ..., r_{n-L+1}\right]^T$: Filter input vector.

$\hat{r}_{n+1} = \mathbf{a}_n^T \mathbf{r}_n$: Filter output with the tap-weight vector $\mathbf{a}_n$.

r_{n+1}: Desired response.

$\alpha_{n+1} = r_{n+1} - \hat{r}_{n+1} = r_{n+1} - \mathbf{a}_n^T \mathbf{r}_n$: The regularized prediction error obtained at the output of the regularized estimator. Note that α_n must not be confused with the ESN bank's prediction error, e_n.

$\beta_n = \hat{r}_{n+1} - \hat{r}_n = \mathbf{a}_n^T \mathbf{r}_n - \mathbf{a}_{n-1}^T \mathbf{r}_{n-1}$: Difference between two consecutive estimates.

The cost function (J) associated with the adaptive estimator is to be minimized with respect to the weight vector, $\mathbf{a}_n$:

$$\begin{aligned} J &= J_s + \lambda J_{reg} \\ &= \mathbb{E}\left[\alpha_{n+1}^2\right] + \lambda \|\beta_{n+1}\|^2 \\ &= \mathbb{E}\left[(r_{n+1} - \hat{r}_{n+1})^2\right] + \lambda \left\|\hat{r}_{n+1} - \hat{r}_n\right\|^2 \\ &= \mathbb{E}\left[\left(r_{n+1} - \mathbf{a}_n^T \mathbf{r}_n\right)^2\right] + \lambda \left\|\mathbf{a}_n^T \mathbf{r}_n - \mathbf{a}_{n-1}^T \mathbf{r}_{n-1}\right\|^2 . \end{aligned} \tag{3.17}$$

J_s is the standard error term which minimizes the mean square deviation between r_{n+1} and its predictor $\hat{r}_{n+1} = \sum_{l=1}^{L} a_l r_{n-l+1} = \mathbf{a}_n^T \mathbf{r}_n$. J_{reg} is the regularizing term, which penalizes the sudden variations in the estimates as a consequence of the underlying smoothness assumption of the driving-force signal. λ is the regularization constant that provides a balance between the standard error term and the regularization term. The regularized estimator is trained on the LMS algorithm iterating the estimator's weights in the inverse direction of the gradient scaled by an adequate step-size parameter at each time instant. Following the methodology presented by Kushner and Yang (1995), we update not only the tap-weights, but also the step-size parameter, μ. Leaving the derivations to the Appendix A.1, the regularized LMS equations with adaptive gain are obtained as given below.

$$\begin{aligned}
\mathbf{a}_{n+1} &= \mathbf{a}_n + \mu_n \left[\alpha_{n+1} - \lambda\beta_{n+1}\right] \mathbf{r}_n \\
\mu_{n+1} &= \mu_n + \gamma \left[\alpha_{n+1} - \lambda\beta_{n+1}\right] \Psi_n^T \mathbf{r}_n \\
\Psi_{n+1} &= \left[\mathbf{I} - (1+\lambda)\,\mu_n \mathbf{r}_n^T \mathbf{r}_n\right] \Psi_n + \left[\alpha_{n+1} - \lambda\beta_{n+1}\right] \mathbf{r}_n
\end{aligned} \tag{3.18}$$

where $\mathbf{I}$ denotes an $L \times L$ identity matrix, and the $L \times 1$ vector $\Psi_n \triangleq \nabla_{\mu_n} \mathbf{a}_n$ facilitates the update of the step-size parameter, μ_n.

We anticipate that the resulting estimator will be capable of reconstructing both slowly and rapidly varying driving-forces from arbitrary nonlinear dynamics thanks to the power and fast convergence of the ESNs, and the robustness of the regularized LMS adaptation algorithm. Nevertheless, the nature of the estimation task may place some limitations on the algorithm's performance. For instance, if the partial derivative in (3.6) has a (possibly nonlinear) dependence on the driving-force u_n, arguably a more challenging estimation scenario is posed. It can be also expected that estimating the rapidly changing driving-forces would be ordinarily more difficult than estimating those forces that exhibit slower variations. The innate degree of difficulty of a particular estimation task can be evaluated in a principled way by the Posterior Cramer-Rao Lower Bound (PCRB). Then comparing the algorithm's performance with the PCRB, we can reach some rigorous conclusions on the algorithm's performance under varying conditions.

Before concluding this section, we summarize the operation of the ADFE in an algorithmic form as given in Algorithm 3. Lines between 1 and 7 address the ESN operations. Line 8 describes the noise-smoother, and the operations between lines 9 and 14 define the regularized adaptive estimator.

Algorithm 3 THE ADAPTIVE DRIVING-FORCE ESTIMATOR

```
Require: Observables from the environment; $y_n, n = 1, ..., \mathcal{T}$
Ensure: Driving-force estimates; $\hat{r}_n, n = 1, ..., \mathcal{T}$
    Initialize the RLS algorithm by setting $\mathbf{P}_0^{(i)} = \delta\mathbf{I}$ and $\mathbf{w}_0^{out,(i)} = \mathbf{0}$ with $\delta$ small.
    $for\ (n = 0; t < \mathcal{T}; n++)\ \{$
        $for\ (i = 1; i \leq V; i++)\ \{$
1:  $\mathbf{s}^{(i)}_{n+1} = \tanh\left(\mathbf{W}^{(i)}\mathbf{s}^{(i)}_n + \mathbf{w}^{in,(i)}y_n + \phi^{(i)}_{n+1}\right)$
2:  $z^{(i)}_{n+1} = \left(\mathbf{w}^{out,(i)}_n\right)^T \mathbf{s}^{(i)}_{n+1}$
3:  $e^{(i)}_{n+1} = y_{n+1} - z^{(i)}_{n+1}$
4:  $\mathbf{k}^{(i)}_{n+1} = \dfrac{\lambda^{-1}_{RLS}\mathbf{P}^{(i)}_n\mathbf{s}^{(i)}_{n+1}}{1 + \lambda^{-1}_{RLS}\left(\mathbf{s}^{(i)}_{n+1}\right)^T \mathbf{P}^{(i)}_n\mathbf{s}^{(i)}_{n+1}}$
5:  $\mathbf{w}^{out,(i)}_{n+1} = \mathbf{w}^{out,(i)}_n + \mathbf{k}^{(i)}_{n+1}e^{(i)}_{n+1}$
6:  $\mathbf{P}^{(i)}_{n+1} = \lambda^{-1}_{RLS}\mathbf{P}^{(i)}_n - \lambda^{-1}_{RLS}\mathbf{k}^{(i)}_{n+1}\left(\mathbf{s}^{(i)}_{n+1}\right)^T \mathbf{P}^{(i)}_n$
        $\}$
7:  $e_{n+1} = (1/V)\sum_{i=1}^{V} e^{(i)}_{n+1}$
8:  $r_{n+1} = (1 - \Gamma)\,e_{n+1} + \Gamma r_n$
9:  $\hat{r}_{n+1} = \mathbf{a}_n^T\mathbf{r}_n$  $\{$where $\mathbf{a}_n \triangleq \left[a_n^{(1)}, \ldots, a_n^{(L)}\right]^T$ and $\mathbf{r}_n \triangleq [r_n, \ldots, r_{n-L+1}]^T\}$
10: $\alpha_{n+1} = r_{n+1} - \hat{r}_{n+1}$
11: $\beta_{n+1} = \hat{r}_{n+1} - \hat{r}_n$
12: $\mathbf{a}_{n+1} = \mathbf{a}_n + \mu_n\left[\alpha_{n+1} - \lambda\beta_{n+1}\right]\mathbf{r}_n$
13: $\mu_{n+1} = \mu_n + \gamma\left[\alpha_{n+1} - \lambda\beta_{n+1}\right]\Psi_n^T\mathbf{r}_n$
14: $\Psi_{n+1} = \left[\mathbf{I} - (1+\lambda)\mu_n\mathbf{r}_n\mathbf{r}_n^T\right]\Psi_n + \left[\alpha_{n+1} - \lambda\beta_{n+1}\right]\mathbf{r}_n$
    $\}$
```

3.1.3 PCRB

Error lower bounds provide performance limitations for the estimation algorithms, and also allow several methods to be compared against a reference. The Posterior Cramer-Rao Lower Bound (PCRB) is applicable to the estimation of dynamic parameters. Van Trees (2001) derived the PCRB in for the estimation of time-varying parameters which are interpreted as stochastic processes. Tichavsky *et al.* (1998) specified the PCRB for the generalized Markovian nonlinear systems, which is summarized below. Let y represent a sample of measured data, let Θ be an $(\mathsf{P}+1)$ - dimensional estimated random parameter, let $p(y,\Theta)$ be the joint probability density of the pair (y,Θ) and let $\hat{\Theta}$ be an estimate of Θ. PCRB on the estimation error has the form

$$P \triangleq \mathbb{E}\left[\left(\hat{\Theta}-\Theta\right)\left(\hat{\Theta}-\Theta\right)^{T}\right] \geq \mathbf{J}^{-1}, \tag{3.19}$$

where $\mathbf{J}$ is the $(\mathsf{P}+1)\times(\mathsf{P}+1)$ (Fisher) information matrix with the elements

$$J_{ij}=\mathbb{E}\left[-\frac{\partial^{2}\log p(y,\Theta)}{\partial\Theta_{i}\Theta_{j}}\right] \quad i,j=1,...,\mathsf{P}+1. \tag{3.20}$$

Let us use the following notation for the first and second-order partial derivatives respectively, $\nabla_{\Theta}=\left[\frac{\partial}{\partial\Theta_{1}},\ldots,\frac{\partial}{\partial\Theta_{k+1}}\right]^{T}$; $\Delta_{\Xi}^{\Theta}=\nabla_{\Xi}\nabla_{\Theta}^{T}$. Using the relation $p(y,\Theta)=p(y|\Theta)\,p(y)$, and keeping in mind that $p(y)$ does not depend on Θ, we reach

$$J=\mathbb{E}\left[-\Delta_{\Theta}^{\Theta}\log p(y|\Theta)\right]. \tag{3.21}$$

Now let us consider the general nonlinear filtering problem,

$$\begin{aligned}\mathbf{x}_{n+1}^{(1)}&=g_{1}(\mathbf{x}_{n})+\omega_{n}^{(1)} \qquad (3.22)\\ \mathbf{x}_{n+1}^{(2)}&=g_{2}(\mathbf{x}_{n})+\omega_{n}^{(2)}\\ y_{n+1}&=h(\mathbf{x}_{n+1})+v_{n+1},\end{aligned}$$

where we have split the state transition into two parts. Note that dividing the state equation into two sub-equations enables us to incorporate the evolution of the driving-force into the system model as an augmented state. Note that the RBPF and the EM-PF have been described on such an augmented system as given in (2.21). The state variables $\mathbf{x}_n^{(1)}$ and $\mathbf{x}_n^{(2)}$ in (3.22) are defined as $(\mathsf{P}_1 + 1) \times 1$ and $\mathsf{P}_2 \times 1$ vectors respectively, such that $\mathsf{P}_1 + \mathsf{P}_2 = \mathsf{P}$. The $(\mathsf{P}+1)\times 1$ stacked state vector is given as $\mathbf{x}_n \triangleq [(\mathbf{x}_n^{(1)})^T, (\mathbf{x}_n^{(2)})^T]^T$. The mappings $g_1(.)$ and $g_2(.)$ represent arbitrary nonlinearities, and additive noises $\omega_n^{(1)}$ and $\omega_n^{(2)}$ respectively are white. $h(.)$ is the measurement mapping. Let us concatenate the state vectors and the measurements up to time n in respective vectors such that $\mathbf{X}_n \triangleq [\mathbf{x}_0^T, \ldots, \mathbf{x}_n^T]^T$ and $\mathbf{y}_n \triangleq [y_0, \ldots, y_n]^T$ respectively. Equations in (3.22) and $p(\mathbf{x}_0)$ define the joint pdf of $\mathbf{X}_n$ and $\mathbf{y}_n$ such that,

$$p(\mathbf{X}_n, \mathbf{y}_n) = p(\mathbf{x}_0) \prod_{j=1}^{n} p(y_j|\mathbf{x}_j) \prod_{k=1}^{n} p(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{3.23}$$

Let $\mathbf{J}(\mathbf{X}_n)$ denote the entire $n(\mathsf{P}+1) \times n(\mathsf{P}+1)$ information matrix of the vector $\mathbf{X}_n$ derived from the above joint density. We are interested in determining the information submatrix for estimating $\mathbf{x}_n$, which is denoted $\mathbf{J}_n$, that is given as the inverse of the $(\mathsf{P}+1) \times (\mathsf{P}+1)$ right-lower block of $\mathbf{J}(\mathbf{X}_n)$. The matrix $\mathbf{J}_n^{-1}$ will provide a lower bound on the mean square error for estimating $\mathbf{x}_n$. Decomposing $\mathbf{X}_n$ in time as $\mathbf{X}_n \triangleq \left[\mathbf{X}_{n-1}^T, \mathbf{x}_n^T\right]^T$, we obtain the following recursion for the sequence $\{\mathbf{J}_n\}$ of posterior information submatrices (Tichavsky *et al.*, 1998),

$$\mathbf{J}_{n+1} = \mathbf{G}_n^{22} - \mathbf{G}_n^{21} \left(J_n + \mathbf{G}_n^{11}\right)^{-1} \mathbf{G}_n^{12}. \tag{3.24}$$

The initial information submatrix can be calculated from the a priori pdf, $\mathbf{J}_0 = \mathbb{E}\left[-\Delta_{x_0}^{x_0} \log p\left(\mathbf{x}_0\right)\right]$. Note that $p\left(\mathbf{x}_{n+1}|\mathbf{x}_n\right) = p\left(\mathbf{x}_{n+1}^{(1)}|\mathbf{x}_n\right) p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right)$ is determined from the first and second equations in (3.22). The matrices $\mathbf{G}_n^{ij}$ are given as

$$\begin{aligned}
\mathbf{G}_n^{11} &= \mathbb{E}\left[-\Delta_{\mathbf{x}_n}^{\mathbf{x}_n} \log p\left(\mathbf{x}_{n+1}^{(1)}|\mathbf{x}_n\right)\right] + \mathbb{E}\left[-\Delta_{\mathbf{x}_n}^{\mathbf{x}_n} \log p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right)\right] \\
\mathbf{G}_n^{12} &= \mathbb{E}\left[-\Delta_{\mathbf{x}_{n+1}}^{\mathbf{x}_n} \log p\left(\mathbf{x}_{n+1}^{(1)}|\mathbf{x}_n\right)\right] + \mathbb{E}\left[-\Delta_{\mathbf{x}_{n+1}}^{\mathbf{x}_n} \log p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right)\right] \\
\mathbf{G}_n^{21} &= \left[\mathbf{G}_n^{12}\right]^T \\
\mathbf{G}_n^{22} &= \mathbb{E}\left[-\Delta_{\mathbf{X}_{n+1}}^{\mathbf{X}_{n+1}} \log p\left(\mathbf{x}_{n+1}^{(1)}|\mathbf{x}_n\right)\right] + \mathbb{E}\left[-\Delta_{\mathbf{X}_{n+1}}^{\mathbf{X}_{n+1}} \log p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right)\right] \\
&\quad + \mathbb{E}\left[-\Delta_{\mathbf{X}_{n+1}}^{\mathbf{X}_{n+1}} \log p\left(y_{n+1}|\mathbf{x}_{n+1}\right)\right].
\end{aligned} \tag{3.25}$$

3.1.4 Controlled Experiments

Experiments on Chaotic Maps

The estimator's performance with regard to the chaotic dynamics will be illustrated on two different systems. The first system under consideration is a simple, yet extremely rich Markovian example, the logistic map as described in (3.26):

$$\begin{aligned}
x_{n+1} &= u_n x_n \left(1 - x_n\right) \\
y_{n+1} &= x_{n+1} + v_{n+1},
\end{aligned} \tag{3.26}$$

where the chaotic state x_n is linear in the driving-force, u_n. y_n is the measured sample at time n. Note the logistic map has been introduced in (2.5) in Section 2.1 with a constant bifurcation factor, r_o. The logistic map can be driven in different regions of operation by controlling the value of r_o. (3.26) differs from (2.5) in that the constant bifurcation scalar r_o in (2.5) is replaced by a dynamic bifurcation process u_n in (3.26), which plays the role of the driving-force that we wish to estimate. The second system under study is the Moran-Ricker map, which distinguishes itself from the logistic map mainly in that the chaotic state x_n is related to the driving-force u_n by a nonlinear recursion as given in (3.27),

$$\begin{aligned} x_{n+1} &= x_n \exp\left[\kappa\left(1 - x_n/u_n\right)\right] \qquad (3.27) \\ y_{n+1} &= x_{n+1} + v_{n+1} \end{aligned}$$

where the map can be operated in a chaotic regime by tuning the constant κ adequately.

Note that the sources of uncertainty in the chaotic systems in (3.26) and (3.27) are threefold. Hence, the performance of the algorithm is studied with an emphasis on these three points. In order to quantify the impact of all three factors, we comment on how those points pose particular challenges, and how each of them manifests itself in the derivation of the PCRB:

- The first source of uncertainty is due to the initial state, x_0. To quantify the impact of varying initial conditions in a differentiable dynamic system, it is natural to consider the derivatives $\dfrac{\partial x_n}{\partial x_0}$. These derivatives can be obtained via the chain rule. To study the long term behavior, let us consider the Lyapunov exponent $\Gamma = \lim_{n\to\infty} \dfrac{1}{n} \log \left[\dfrac{\partial x_n}{\partial x_0}\right]$. The intuitive approximation $\exp(n\Gamma) \approx \dfrac{\partial x_n}{\partial x_0}$ suggests that for large Γ, small deviations in the initial conditions lead to large deviations in x_n (Berliner, 1991). As a result, different initializations of a chaotic map may lead to different horizons of predictability. The impact of x_0 is taken into account in (A.7) and (A.12) for the respective derivation and calculation of the PCRB in Appendix A.2.

- The impact of the partial derivative in (3.6), as mentioned in the last paragraph of Section 3.1.2, manifests itself in the PCRB as given in (A.13) and in (A.14) in Appendix A.2. Specifically, it follows from (A.13) that

$$\frac{\partial f\left(y_n, u_n\right)}{\partial u_n} = \eta_n = x_n\left(1 - x_n\right) \tag{3.28}$$

for the logistic map, and it is deduced from (A.14) that

$$\frac{\partial f\left(y_n, u_n\right)}{\partial u_n} = \eta_n = \kappa\frac{x_n^2}{u_n^2}\exp\left[\kappa\left(1 - x_n/u_n\right)\right] \tag{3.29}$$

for the Moran-Ricker map. Note that η_n in (3.28) does not depend on u_n, whereas η_n in (3.29) has a severely nonlinear relation with u_n for the Moran-Ricker map. This observation suggests that estimating a driving-force from the Moran-Ricker map will be a more challenging problem than from the logistic map. As will be presented in the experiments, the PCRB will serve as a quantitative measure of this comparison.

- Another source of uncertainty is the additive measurement noise denoted by v_n. Examining (3.3) and (3.4) which describe the derivation of the hybrid mapping $f(.)$, it can be seen that the effect of the measurement noise can be greatly amplified when the state mapping $g(.)$ is nonlinear, which may severely impair the neural network's prediction capability. This is due to the fact that the underlying smoothness assumption of the hybrid mapping $f(.)$ could be violated in such cases. The nonlinearity of $g(.)$ is indeed the case for both examples under consideration. The effect of measurement noise on the PCRB can be observed following from (A.11) in Appendix A.2.

The following two comments are in order prior to the presentation of the experimental setup and the results:

1. It should be noted that we perform a sanity check on the prediction error of the ESN bank. Specifically, we monitor the ESN bank's training error to make sure that the prediction error remains bounded.

2. Note that the PCRB is deriven on the availability of the full model of the underlying system. The proposed ADFE however does not assume the availability of the model knowledge. Therefore, the PCRB is interpreted as a loose approximation to the actual lower bound for the experiments in sequel.

In light of the points stressed above, and following from the detailed derivation of the PCRB given in Appendix A.2, we are ready to proceed to the controlled experiments.

We study the reconstruction of two driving-forces that differ in their respective time-scales. In particular, the following amplitude-modulated (AM) signal will be considered as an example of a rapidly varying driving-force;

$$u_n^{(1)} = r_o + A_c \left(1 + k_a \cos\left(2\pi f_m n T_s\right)\right) \cos\left(2\pi f_c n T_s\right), \tag{3.30}$$

where A_c is the amplitude, k_a is the modulation index, f_m is the message frequency, T_s is the sampling period, and f_c is the carrier frequency. The carrier frequency dictates the global time-variation of $u_n^{(1)}$, whereas the message frequency determines the shape of its envelope. As a result, this driving-force changes in two time-scales. $u_n^{(1)}$ will be realized with the following parameters: For logistic map; $r_o = 3.8$, $A_c = 0.125$, $k_a = 0.7$. These values ensure that the logistic map exhibits a chaotic behavior. $f_m = 10$ Hz, $f_c = 100$ Hz, $f_s = 1/T_s = 2000$ Hz are selected, which results in a rapidly varying signal. For the Moran-Ricker map, $r_o = 1$, $A_c = 0.6$, and the other parameters are the same. $T = 2000$ data points will be drawn from the logistic map and the Moran-Ricker map for the reconstruction of $u_n^{(1)}$.

To test the algorithm's performance on a slowly varying driving-force, the following signal will be considered,

$$u_n^{(2)} = r_o - A\cos\left(\frac{2\pi n}{50}\right) e^{-n/50}, \tag{3.31}$$

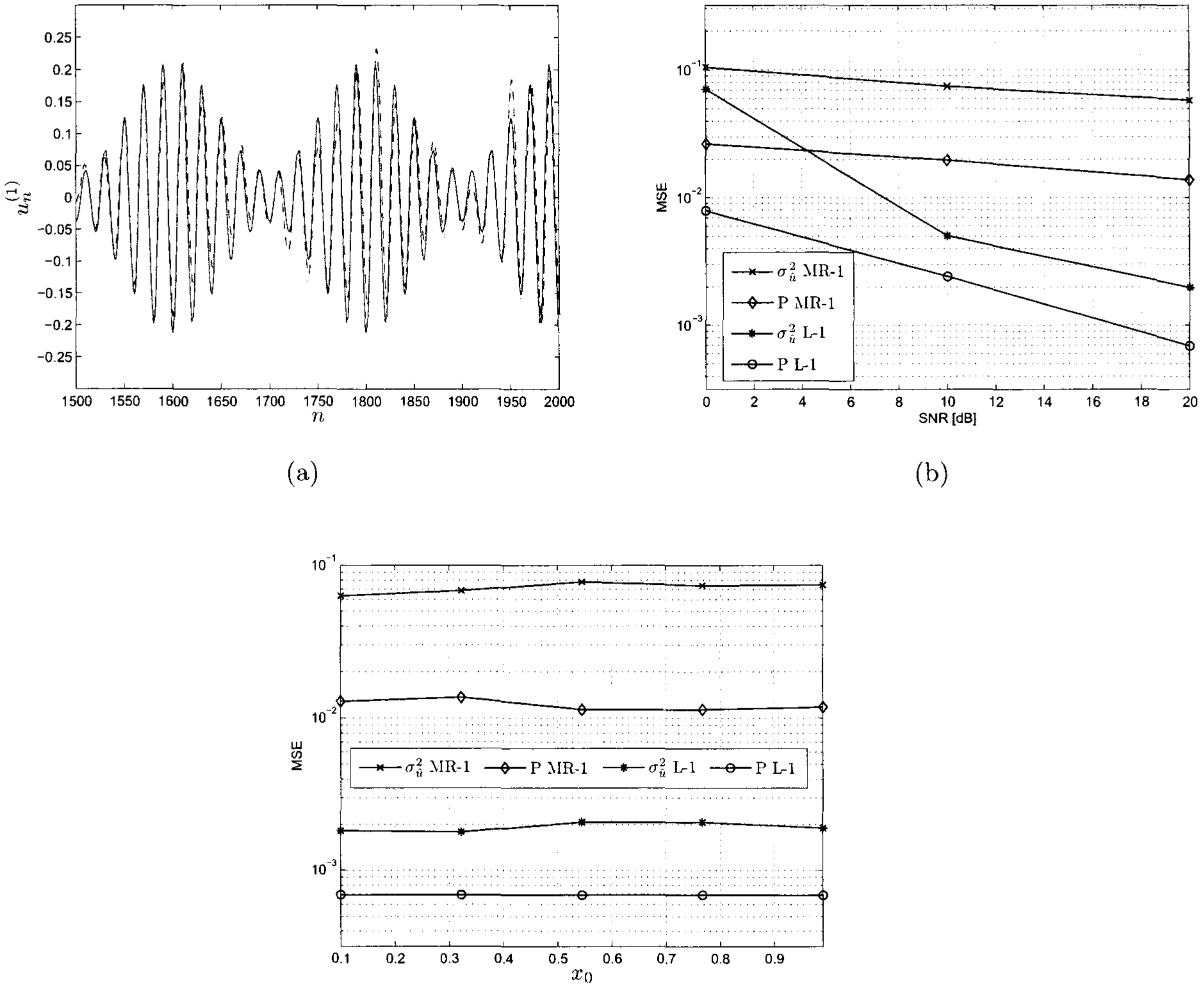
which was previously studied by Szeliga *et al.* (2003a,b); Verdes *et al.* (2006) with the following parameters for the logistic map: $r_o = 3.8$, $A = 0.045$. These values ensure that the logistic map stays in the nontrivial chaotic regime. The same signal will be simulated with the following parameters in the Moran-Ricker map: $r_o = 1$, $A = 0.9$. $T = 100$ data points will be drawn from the logistic map and the Moran-Ricker map for the reconstruction of $u_n^{(2)}$. For the Moran-Ricker map, $\kappa = 3.7$ is set for the reconstruction of both $u_n^{(1)}$ and $u_n^{(2)}$.

In light of the experimental results presented in Fig.3.2, the ESNs are configured with the following parameters, which are fixed throughout the upcoming experiments also: $\varsigma = 0.1$, $\rho = 0.1$, $N = 30$. For $u_n^{(1)}$, we have a relatively large dataset, (i.e. $T = 2000$). Therefore, we can set the number of estimator taps to $L = 100$. Due to the fact that we draw only 100 data points for $u_n^{(2)}$, which is relatively small, we set the adaptive estimator length at $L = 10$ for $u_n^{(2)}$. We have observed that as long as the step-size parameter is initialized with a small number (e.g. $0 \leq \mu_0 \leq 10^{-3}$), the algorithm's performance is insensitive to the selection of μ_0. For this and the remaining experiments, the initial step-size is fixed at $\mu_0 = 10^{-5}$. The regularization constant is set $\lambda = 1$ for all experiments. The sensitivity of the algorithm to the selection of λ will be studied in a unified manner for all applications at the end of Section 3.1.5.

The algorithm's performance for the reconstruction of the rapidly varying AM force is presented in Fig.3.3. In Fig.3.3(a), an example run of the algorithm is displayed. For visual convenience, only the last 500 samples are plotted. In Fig.3.3(b), performance of the algorithm is shown for varying degrees of measurement noise in terms of Signal-to-Noise Ratio (SNR). The SNR is measured as $SNR = 10\log_{10}\left(\frac{P_u}{\sigma_v^2}\right)$, where P_u is the power of the driving-force signal, and σ_v^2 is the variance of measurement noise. For each SNR level, the corresponding values of the variance of estimation error $\sigma_{\tilde{u}}^2$ and the PCRB are shown. The first 300 samples are removed from the calculations so as to discard the effect of transient dynamics. Comparing the PCRB for the MR-map and the PCRB for logistic map in Fig.3.3(b), we can deduce that the MR-map indeed poses a harder problem than the logistic map. The algorithm's performance on the MR-map follows the PCRB almost at the same distance for all SNR levels. For the logistic map however, the algorithm performs well for moderate to high SNR (i.e. $10 - 20$ dB), whereas the performance is largely degraded for 0 dB. Also, the PCRB for logistic map declines remarkably faster than the PCRB for the MR-map for increasing SNR. The reason for these two observations is that the driving-force is carried in the amplitude content of the logistic map, in which case the white noise is directly added on the driving-force. For the MR-map however, the driving-signal is in the phase content, and therefore less affected by the additive noise. As a result, there is still room for improvement for the estimation of rapidly varying forces for high levels of additive noise. In Fig.3.3(c), we plot the variance of estimation error and the PCRB for different initial states, x_0. Although the MSE of the algorithm exhibits small fluctuations for varying x_0, we can conclude that the algorithm is reasonably insensitive to the selection of the initial conditions.

The results of experiments for the extraction of slowly varying force $u_n^{(2)}$ are presented next in Fig.3.4. In Fig.3.4(a), 100 independent realizations of the algorithm are illustrated for the logistic map example. It is seen that the algorithm keeps track of the driving-force after the transient period is completed. In Fig.3.4(b), the noise performance of the algorithm is shown. First 10 samples are discarded from the calculation of MSE to discard the transient dynamics. Comparing Fig.3.4(b) and Fig.3.3(b), similar conclusions can be drawn regarding the sensitivity of the method to the additive noise. However one interesting point is that for the logistic map example, the additive noise is less harmful for the estimation of $u_n^{(2)}$ than it was for $u_n^{(1)}$. The reason is that the adaptive filter operates as a low-pass filter with an adaptive bandwidth. Therefore, for the low frequencies (i.e. slowly varying forces), the filter bandwidth is narrow enough to effectively smooth out most of the noise content. The converse is true for the rapidly varying forces. Finally, in Fig.3.4(c), the sensitivity to the initial state is illustrated. It can be concluded that the algorithm is fairly insensitive to the initial conditions for the estimation of $u_n^{(2)}$ as it was the case for $u_n^{(1)}$.

Note that the driving-force is injected in different strengths in chaotic systems studied above so as to keep the maps in the chaotic regime without collapsing them

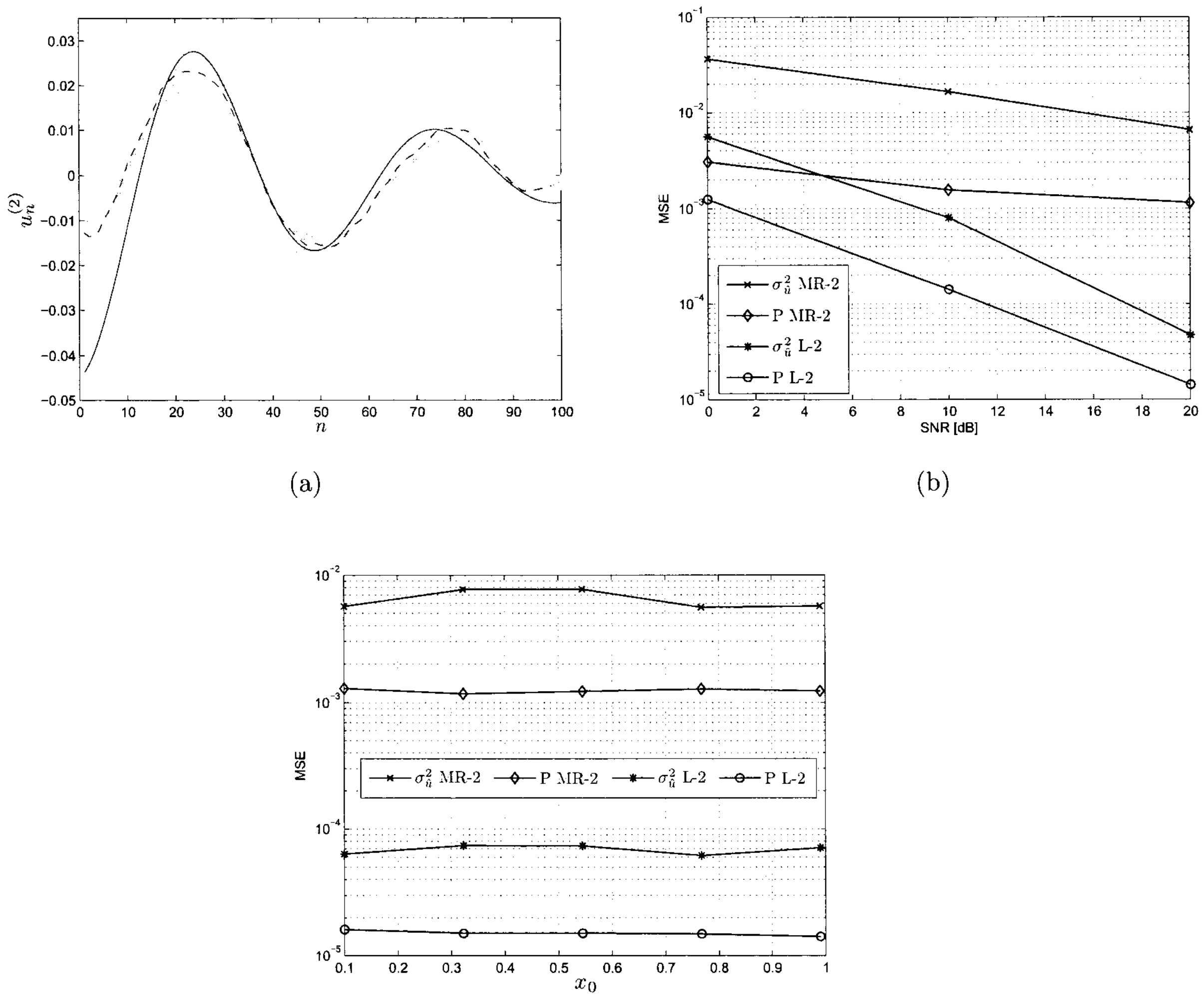


(a)

(b)

(c)

Figure 3.3: Experimental results for the reconstruction of the rapidly varying AM driving-force $u_n^{(1)}$ given in (3.30) from Moran-Ricker and Logistic maps.
(a) Sequential driving-force estimation performed on the AM-force driven logistic map. Solid line is the original driving-force $u_n^{(1)}$. Dashed curve is the average reconstruction. The shaded area refers to the deviation of 100 independent reconstructions from the average reconstruction. $x_0 = 0.42$, SNR=20 dB.
(b) Performance of the algorithm for varying SNR. 'MR-1' and 'L-1' stand for the Moran-Ricker map and the logistic map perturbed by $u_n^{(1)}$ respectively. $\sigma_{\tilde{u}}^2$ denotes the time-averaged variance of estimation error, P denotes the PCRB. $x_0 = 0.42$.
(c) Performance of the algorithm for varying initial state, x_0. 'MR-1' and 'L-1' stand for the Moran-Ricker map and the logistic map perturbed by $u_n^{(1)}$ respectively. SNR=20 dB.



(a)

(b)

(c)

Figure 3.4: Experimental results for the reconstruction of the slowly varying $u_n^{(2)}$ given in (3.31) from Moran-Ricker and Logistic maps.

(a) Sequential driving-force estimation performed on the slowly-driven logistic map. Solid line is the original driving-force $u_n^{(2)}$. Dashed curve is the average reconstruction. The shaded area refers to the deviation of 100 independent reconstructions from the average reconstruction. $x_0 = 0.42$, SNR=20 dB.

(b) Performance of the algorithm for varying SNR. 'MR-2' and 'L-2' stand for the Moran-Ricker map and the logistic map perturbed by $u_n^{(2)}$ respectively. $\sigma_{\hat{u}}^2$ denotes the time-averaged variance of estimation error, P denotes the PCRB. $x_0 = 0.42$. dB.

(c) Performance of the algorithm for varying initial state, x_0. 'MR-2' and 'L-2' stand for the Moran-Ricker map and the logistic map perturbed by $u_n^{(2)}$ respectively. SNR=20 dB.

to the unstable regions. Therefore, we also study the normalized mean square error (NMSE) as a global metric for a unified performance assessment. The results are presented Table 3.1 where SNR=20 dB and $x_0 = 0.42$ are fixed. The NMSE is calculated as $\sigma_{\tilde{u}}^2/P_u$, where $\sigma_{\tilde{u}}^2$ is the variance of estimation error, and P_u is the power (or variance) of the driving force. Comparing L-1 with MR-1 and L-2 with MR-2 in Table 3.1, it is clear that better results are obtained for the logistic map than the MR-map. This observation is in agreement with the results presented in (Verdes *et al.*, 2001). Nevertheless, the normalized error for the MR map is still smaller than 0.1 as well. Another comparison between L-1 and L-2, and MR-1 and MR-2 reveals that both slowly and rapidly varying forces can be reconstructed by the algorithm up to a similar level of accuracy for reasonably high SNR.

Table 3.1: NMSE performance over all experiments on the chaotic maps. L-1 and MR-1 denote the logistic map and Moran-Ricker map perturbed by $u_n^{(1)}$ in (3.30) respectively. L-2 and MR-2 denote the logistic map and Moran-Ricker map perturbed by $u_n^{(2)}$ in (3.31) respectively.

	L-1	MR-1	L-2	MR-2
NMSE	0.068	0.09	0.077	0.093

Before concluding this subsection, we finally present some comparison between the results obtained herein and those that were recorded by Verdes *et al.* (2006, Table 1). The authors of (Verdes *et al.*, 2006) quantified the additive noise level in terms of the ratio of standard deviation of the noise to the standard deviation of the driving-signal. We transform those numbers to signal-to-noise ratios in dB, which is the standard power ratio unit throughout this chapter. The results are presented in Table 3.2. Clearly, the method of Verdes *et al.* (2006) provides an impressive performance for the noise-free case (i.e. SNR=∞), and for very low noise levels (i.e. SNR=40 dB). The proposed approach however distinguishes itself in its much greater stability against the increasing levels of additive noise exemplified by the results for 26 and 20 dB.

Table 3.2: NMSE performance comparison on L-2 experiment for varying SNR between the proposed approach and (Verdes *et al.*, 2006). $T = 100$.

SNR [dB]	∞	40	26	20
NMSE in (Verdes *et al.*, 2006)	0.017	0.013	0.088	0.191
NMSE herein	0.068	0.067	0.071	0.077

Experiments on a Nonlinear Stochastic System

In the preceding subsection, we have examined the algorithm's performance on two chaotic systems, whose state equations had zero process noise. The uncertainty in the state models was rooted from the deterministic chaos. In this set of experiments, we study another nonlinear dynamic system as given in (3.32), where both the state and measurement models are stochastic. This will allow us to judge the algorithm's performance on such scenarios where the system dynamics can carry varying degrees of stochastic uncertainty represented by the process noise. We study the following system,

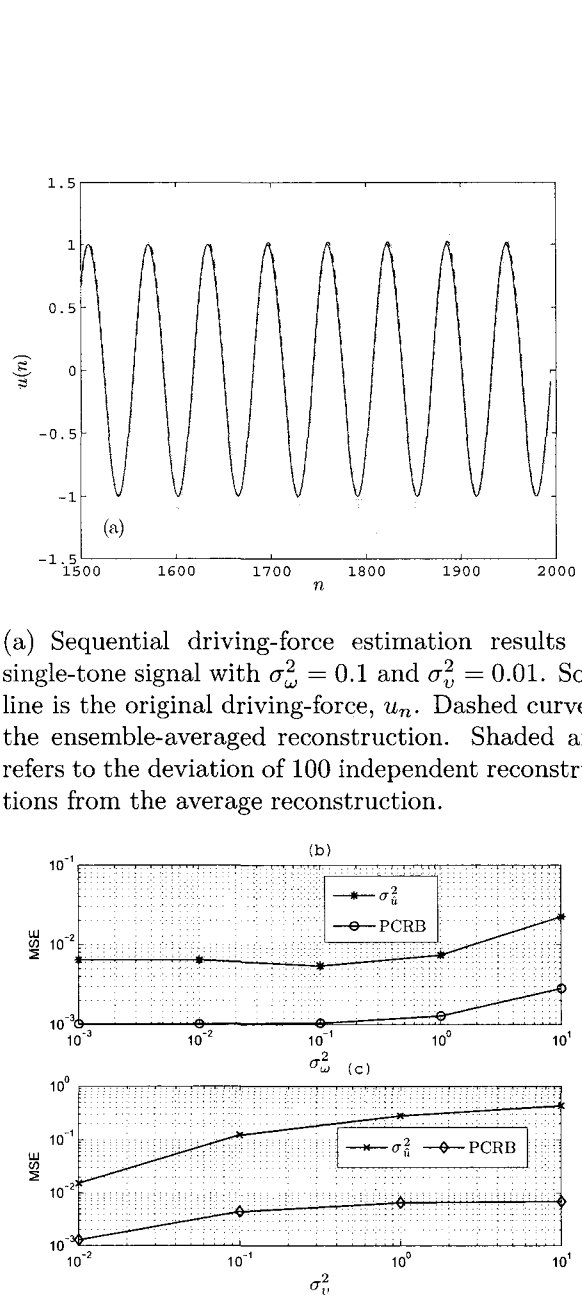
$$\begin{aligned} x_{n+1} &= 0.5x_n + \frac{25x_n}{1+x_n^2} + u_n + \omega_{n+1} \\ y_{n+1} &= \frac{x_{n+1}}{20} + v_{n+1}, \end{aligned} \tag{3.32}$$

which was originally studied by Gordon *et al.* (1993) as a case of severe nonlinearity for Bayesian sequential nonlinear state estimation problem. ω_n denotes the process noise. Our problem is, once again, to estimate u_n without the knowledge of the system. The results will be also judged quantitatively with a reference to the PCRB. We skip the derivation of PCRB for this example since it follows from the equations presented in Section 3.1.3 in a rather straightforward manner.

We perturb the system in (3.32) with a single-tone signal, $u_n = \cos(n/10)$. The configurations pertaining to the ESN and the adaptive estimator are the same as in the preceding section for estimating $u_n^{(1)}$.

In Fig.3.5(a), we illustrate the estimates obtained by 100 independent trials. Only the last 500 samples are plotted for easy visualization.

We test the algorithm's performance on varying levels of process noise in Fig.3.5(b). It is observed that the variance of the estimation error follows a similar trend as the PCRB at almost a uniform distance. This observation suggests that the algorithm's performance is fairly insensitive to the uncertainty in the governing body of system dynamics. This is an expected result since the process noise ω_n is introduced as additive noise in the expression of the hybrid system mapping, $f(.)$ in (3.4). A large process noise therefore does not greatly impede the prediction capability of the ESN predictor since the ESN bank approach provides an ensemble averaging mechanism. However, inspecting (3.4) and (3.3) once again, we observe that the measurement noise v_n is located in the argument of the nonlinear state transition represented by $g(.)$. Therefore, similar comments can be suggested regarding the effect of measurement noise as those presented for the chaotic map experiments in the preceding subsection. In particular, increasing levels of the measurement noise is expected to affect the smoothness of $f(.)$, hence the predictability of the observables by the neural networks. The algorithm's performance on varying measurement noise is presented in Fig.3.5(c), where the distance between the variance of estimation error and the PCRB gets larger for increasing measurement noise. In principle, following from Cover's theorem (Cover, 1965), it could be possible to reduce the harming effect of measurement



(a) Sequential driving-force estimation results for single-tone signal with $\sigma_\omega^2 = 0.1$ and $\sigma_v^2 = 0.01$. Solid line is the original driving-force, u_n. Dashed curve is the ensemble-averaged reconstruction. Shaded area refers to the deviation of 100 independent reconstructions from the average reconstruction.

(b) Effect of process noise. The variance of measurement noise is fixed at $\sigma_v^2 = 0.01$. (c) Effect of measurement noise. The variance of the process noise is fixed at $\sigma_\omega^2 = 1$. $\sigma_{\tilde{u}}^2$ denotes the variance of the estimation error.

Figure 3.5: Results of the sequential driving-force estimation algorithm performed on the nonlinear stochastic system in (3.32).

noise by transforming the data into an even higher dimensional space. This can be done by using a nonlinear readout layer for the ESN. The feasibility of this approach will be suggested as part of the future research.

3.1.5 Application To Sunspot Time Series

We finally illustrate the performance of the estimator on some real-life data, the sunspot time series, in which the hidden information exhibits an irregular behavior.

A sunspot is the cooler darker spot appearing on the sun's photosphere whose mechanism is not exactly known. For sunspot time series, the dominating perturbation is the sun's total magnetic flux, hence, the driving-force is one-dimensional. A mathematical model is developed by Solanki *et al.* (2002) to extract the information pertaining to sun's magnetic flux. Lockwood *et al.* (1999) used a different method for estimating sun's magnetic flux. Specifically, a set of measurements called the 'aa' index has been compiled from the measurements of geomagnetic field since 1868 by pairs of magnetometers. An analytical model is always preferable to an algorithmic approach whenever there is enough prior information and physical evidence to construct a rigorous but yet uncomplicated model. That being said, we do not pursue to claim that our method could be comparable to the analytical models. Rather, following the work presented by Verdes *et al.* (2004), we interpret the work of Solanki *et al.* (2002) and Lockwood *et al.* (1999) as the true models, and refer to the results therein to judge the performance of our algorithm by visual inspection.

We used the international annual sunspot numbers available from the year 1700 to 2008 (NGDC, 2009). The ESN configuration used in this experiment is the same as in the previous experiments.

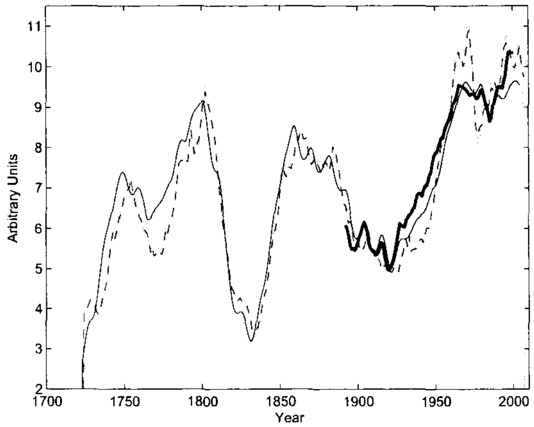


Figure 3.6: Reconstruction of the Sun's total magnetic flux. Thin solid lines: construction by the analytical study by Solanki *et al.* (2002). Thick solid lines: Results of the study by Lockwood *et al.* (1999) (Available only since year 1868). Dashed lines: Average reconstruction of 100 independent trials obtained by the proposed method from the normalized time series. Shaded area: deviation of 100 independent reconstructions from the average reconstruction of the proposed algorithm.

In Fig.3.6, we show the 20-year running mean of the sun's total magnetic flux estimated by the proposed method as well as the reconstruction obtained by Solanki *et al.* (2002), and Lockwood *et al.* (1999). Although we do not expect to obtain an exact match with the analytical model results, we observe that our method has captured the basic characteristics of the driving-force. Especially the rapid increase in the sun's total magnetic flux in the last century is also visible as seen in Fig.3.6.

Sensitivity to the Selection of the Regularization Parameter

We analyze the algorithm's sensitivity to the selection of the regularization parameter, λ over all the experimental setups studied in the current and preceding sections. As inspected in Fig.3.7, a wide plateau is observed for $10^{-4} < \lambda < 1$ for all experiments, where the algorithm does not heavily depend on the selection of the regularization parameter. For sunspot, MR-1 and MR-2 experiments however, the results suggest that the algorithm's performance can be improved by selecting $\lambda = 100$, $\lambda = 1$ and $\lambda = 10$ respectively. The automated selection of λ will be considered by virtue of the generalized cross validation method in the future.

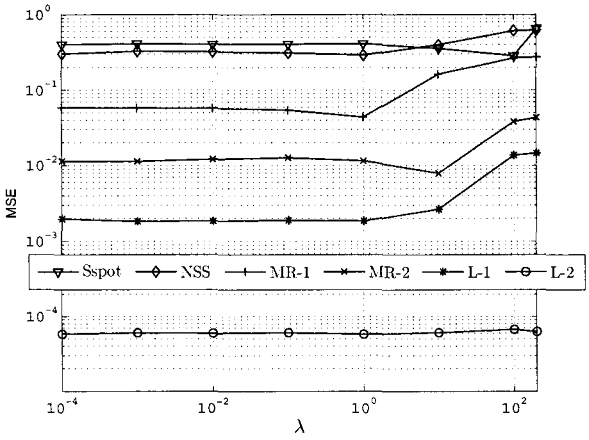


Figure 3.7: Algorithm's sensitivity to the selection of regularization constant, λ. Variance of estimation error for each experiment is plotted against the corresponding values of λ. Following values are set for this experiment: SNR=20dB for the perturbed logistic map, and $\sigma_\omega^2 = 1$, $\sigma_v^2 = 0.1$ for the nonlinear dynamic system. Following labels are used: Sspot: Experiments on sunspot time series; NSS: Experiments on the nonlinear stochastic system in (3.32); MR-1: Moran-Ricker map perturbed by rapidly driven forcing; MR-2: Moran-Ricker map perturbed by the slowly varying force; L-1: Logistic map perturbed by rapidly driven force; L-2: Logistic map perturbed by slowly varying force.

3.1.6 Conclusion

We described a sequential estimation approach building on the prediction error idea for the reconstruction of hidden inputs that perturb nonlinear dynamic systems. The proposed driving-force estimator owes its good performance to the selection of two robust systems; the ESN banks, and the regularized LMS adaptive filtering algorithm.

Both the ESNs and the adaptive filter enjoy a common adaptation property. The ESNs keep learning from the environment by virtue of the continuous update of its free weights, whereas the LMS adaptation algorithm makes it possible to respond to the changes in the environment so as to keep producing regular estimates of the quantity of interest.

The overall adaptation property of the proposed method makes it possible to reconstruct the rapidly varying driving-forces as demonstrated in Section 3.1.4. For that matter, this capability distinguishes the proposed method from the other contributions in the related literature. To this end, as addressed in Section 3.1.4, there is still room for improvement to mitigate the harming effect of additive noise for the estimation of rapidly varying forces. One way to enhance the proposed approach is to modify the design of ESN predictor by replacing the linear readout unit by a nonlinear one as explained at the end of Section 3.1.4.

Slowly varying forces on the other hand can be reconstructed by the algorithm with a similar level of accuracy under varying noise levels and initial conditions. As a matter of fact, we have reported a remarkable improvement in comparison to (Verdes *et al.*, 2006) for the reconstruction of a slowly varying force when the additive noise is an issue (Table 3.2).

For future research, we propose a comparison of the proposed estimator with the Bayesian techniques when prior knowledge on the system equations and noise distributions is partially or fully available. This will enable us to provide solutions to more general situations, as well as to gain more insight into the limitations of the method.

3.2 Toward the Development of a Radar Scene Analyzer for Cognitive Radar

This section is devoted to the presentation of the material published in the article (Güntürkün, 2010b). Since the methodology has already been given in detail in Section 3.1, it is omitted herein. New experimental results and the application of the proposed ADFE to the radar scene analysis problem have been presented in this section.

3.2.1 Introduction

We propose a target signature search algorithm to address the radar scene analyzer in explicit terms for cognitive radar reception in this section. A cognitive radar receiver encompasses two main blocks: The radar scene analyzer (RSA) and the Bayesian target tracker (BTT). The BTT requires prior knowledge of what it is looking for, such as terrain conditions, and the potential targets in that terrain. This information is provided by the RSA with the aid of other external resources. More formally, the statistical information about the environment and the target are central to the realization of the BTT. Modeling the statistics of unknown target returns however is a particularly challenging task. The objective of the proposed RSA structure is to tackle this challenge by extracting useful information about the target from the environment based on the texture modeling of sea clutter. Specifically, we formulate a weak target as an unknown input embedded in the sea surface, whose dynamics are closely coupled by those of the clutter. The mapping that governs the unknown system's dynamics is assumed to be smooth. The observables from the environment are then predicted one-step ahead with a bank of echo state networks (ESN). The unknown target's signature is extracted from the ESN prediction error and then refined in two adaptive filtering stages. Performance of the resulting method is evaluated using the Posterior Cramer-Rao lower bound (PCRB) on some controlled simulations. Finally, the intended application is presented on live recorded sea returns collected by the McMaster IPIX radar. Experiments show that the algorithm can accurately extract the target template from the environment.

Cognitive radar has been proposed as a fully adaptive radar transmission and reception system by Haykin (2006). In cognitive radar, the cognition refers to understanding the environment, and modifying the system parameters accordingly. Both the transmitter and the receiver parameters are updated in the course of time by virtue of learning from the unknown environment, forming a belief on what is learned, and propagating this belief by Bayesian inference. Operating blocks of the cognitive radar are discussed by Haykin (2006). The task associated with the RSA (Haykin, 2006) is to supply statistical information about the clutter and the unknown target. Other prior information (e.g. speed and direction of wind, sea state) provided by external resources can also assist the realization of the BTT. Then the BTT decides whether a particular resolution cell contains a target. The decisions made by the BTT are fed back to the adaptive transmitter, and to the RSA. The parameters of the adaptive transmitter [e.g. transmit waveform, pulse repetition frequency (PRF), polarization] are then updated to focus on the patch of the ocean surface where the target is believed to be located. The range bins labeled by the BTT not to contain a target with high confidence are exploited by the RSA to estimate the time variation of the global clutter dynamics (i.e. the *texture*). A block diagram of the overall system is sketched in Fig.3.8 with the proposed vision of the RSA, which is the focus of this article.

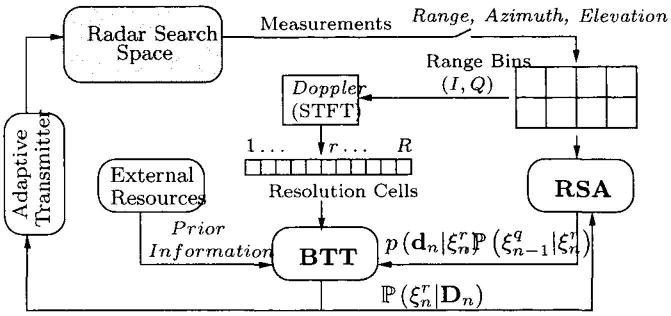

Radar Search Space
Measurements
Range, Azimuth, Elevation
Range Bins
(I, Q)
Doppler (STFT)
$1 \ldots \quad r \ldots \quad R$
Resolution Cells
Adaptive Transmitter
External Resources
Prior Information
BTT
RSA
$p(\mathbf{d}_n|\xi_n^r)\mathbb{P}(\xi_{n-1}^q|\xi_n^r)$
$\mathbb{P}(\xi_n^r|\mathbf{D}_n)$

Figure 3.8: Block diagram of the cognitive radar with the proposed radar scene analyzer.

We propose a realization of the RSA, which employs the proposed adaptive driving-force estimator for extracting hidden-input information from an unknown dynamic environment. To describe the role of the RSA in mathematical terms, let us first briefly describe the sensing routine depicted in Fig.3.8, and then formulate the BTT. The discrete-time measurements from the radar search space are collected in the range bins for each *range, azimuth* and *elevation* dimension. The radar being coherent, the data in the range bins are composed of in-phase (I) and quadrature (Q) components. For the operation of the BTT, sliding windows of the data ($I + jQ$) in each range bin are transformed to the frequency domain by the short-term Fourier transform (STFT). This way, the measurements are indexed into a linear array of resolution cells, which are labeled by $1, ...r, ..., R$. Each resolution cell can now be considered as a 4×1 index vector, whose elements indicate the *range, azimuth, elevation*, and the *Doppler velocity* for a discrete spectral measurement made at time n.[1] The RSA

[1] Let R_1, R_2, R_3 and R_4 denote the number of ranges, azimuth angles, elevation angles, and Doppler velocities respectively. Then the 4 dimensional array of all resolution cells at time n has a cardinality equal to $R = R_1 R_2 R_3 R_4$.

processes the data in the time domain to search for the target signature in a particular range bin of interest. The task of the BTT is then to decide whether a target is contained in any of the R resolution cells at time n exploiting the target signature information provided by the RSA. This is accomplished by calculating the posterior probability, $\mathbb{P}\left(\xi_n^r|\mathbf{D}_n\right)$, where the event of a target being present in the r th resolution cell at time n is denoted by ξ_n^r. The 5–dimensional matrix $\mathbf{D}_n$ is obtained by concatenating the spectral measurements indicated by the arrays of resolution cells from time 1 to n.[2] According to Bayes' theorem, this posterior probability can be expressed as

$$\mathbb{P}\left(\xi_n^r|\mathbf{D}_n\right) = \frac{p\left(\mathbf{D}_n|\xi_n^r\right)\mathbb{P}(\xi_n^r)}{p(\mathbf{D}_n)} \tag{3.33}$$

where $p\left(\mathbf{D}_n|\xi_n^r\right)$ is the likelihood function, denoting the probability density function (pdf) of the measurements given the occurrence of a target in cell r. $\mathbb{P}(\xi_n^r)$ denotes the prior probability of the target appearing in cell r, and $p(\mathbf{D}_n)$ is the normalizing term, which describes the evidence of the measurements. In order to propagate the posterior density in time, let us rearrange the measurement matrix as a concatenation of the measurement vectors ordered in time

$$\mathbf{D}_n = \left[\mathbf{d}_n, \mathbf{d}_{n-1}, ..., \mathbf{d}_1\right] = \left[\mathbf{d}_n, \mathbf{D}_{n-1}\right].$$

The $R \times 1$ vector $\mathbf{d}_i$ denotes the spectral measurements sampled at instants $i = 1, 2, \ldots, n$. By the same token, matrix $\mathbf{D}_{n-1}$ denotes all the spectral measurements made before time n. Rearranging (3.33) and carrying out the necessary recursions,

[2] For instance, the (p, l, k, j, i)th element of the matrix $\mathbf{D}_n$ is a spectral power sample that is measured at the pth range, lth azimuth angle, kth elevation angle, $j$the Doppler velocity, and the ith time instant, where $p = 1, \ldots, R_1$, $l = 1, \ldots, R_2$, $k = 1, \ldots, R_3$, $j = 1, \ldots, R_4$ and $i = 1, \ldots, n$.

one reaches the following simplified expression for the posterior probability

$$\mathbb{P}\left(\xi_n^r|\mathbf{D}_n\right)=\frac{p\left(\mathbf{d}_n|\xi_n^r\right)p\left(\mathbf{D}_{n-1}|\xi_n^r\right)\mathbb{P}(\xi_n^r)}{p(\mathbf{D}_n)} \tag{3.34}$$

where the current frame of measurements $\mathbf{d}_n$ and the set of all previous measurements $\mathbf{D}_{n-1}$ are assumed statistically independent. $p\left(\mathbf{d}_n|\xi_n^r\right)$ is supplied by the RSA. The second term in the numerator in (3.34) is computed by the following recursion:

$$p\left(\mathbf{D}_{n-1}|\xi_n^r\right)=\sum_{q=1}^{R}p\left(\mathbf{d}_{n-1}|\xi_{n-1}^q\right)p\left(\mathbf{D}_{n-2}|\xi_{n-1}^q\right)\mathbb{P}\left(\xi_{n-1}^q|\xi_n^r\right). \tag{3.35}$$

As before, $p\left(\mathbf{d}_{n-1}|\xi_{n-1}^q\right)$ is provided by the RSA. $p\left(\mathbf{D}_{n-2}|\xi_{n-1}^q\right)$ is the one-step delayed version of $p\left(\mathbf{D}_{n-1}|\xi_n^q\right)$ and computed recursively. $\mathbb{P}\left(\xi_{n-1}^q|\xi_n^r\right)$ are the elements of the inverse transition probability matrix. This matrix is also to be received from the RSA and then updated. Consequently, the RSA is expected to generate the following pieces of information for a particular resolution cell:

1. $p\left(\mathbf{d}_n|\xi_n^r\right)$: statistics about the clutter, and/or the target;

2. $\mathbb{P}\left(\xi_{n-1}^q|\xi_n^r\right)$: statistical knowledge on the target's movement.

The Bayesian approach for detection and tracking of targets with known signatures has been analyzed and demonstrated in (Bruno and Moura, 2001a) and (Bruno and Moura, 2001b). In these papers, the target signature is represented by a set of deterministic coefficients, which are assumed known. The signature of a random target is modeled by a conditional pdf in (Bruno and Moura, 2001b, Sec. D). The clutter is modeled as a Gauss-Markov random field (GMRF), whose parameters are determined by maximum likelihood estimation. Based on the GMRF clutter model and the presumed availability of the target signature, the BTT is shown to provide superior performance compared to the Kalman-Bucy filter (Bruno and Moura, 2001a,b). Bakker *et al.* (2002) employ a peak filter before presenting the data to the BTT in order to average-out strong clutter returns, and highlight the target peaks. The output of the peak filter is represented by an $\mathcal{F}$ distribution. The target statistics are modeled as a scaled version of the clutter distribution due to the lack of prior information on the target returns. A major disadvantage of this assumption is that the discriminative power between two cases is weak (Bakker *et al.*, 2002). To our best knowledge, no specific algorithms have been addressed for the realization of the RSA for extracting the signature of a small random target from the environment.

In this chapter, we address this shortcoming by proposing an RSA algorithm which estimates realistic target statistics from the environment by the RSA, that will enable the BTT to operate with more accurate statistics on the target returns.

Before proceeding to the description of the proposed approach, let us stress how the prior knowledge on the target returns is conjectured to be obtained by the ADFE: First, exploiting the texture modeling of sea clutter given by Gini and Greco (2001), we model the background clutter arising from the sea surface as a product between the local scatterers and the global sea waves. Based on this model, we interpret a small random target floating on sea surface as an additional backscatterer, which plays the role of a perturbation input. An intuitive justification of this interpretation lies in the fact that the movement of a small surface target is also dictated by the global sea dynamics in a similar manner to the local backscatterers. Based on the resulting target+clutter for a small surface target, we realize the ADFE as described in 3.1 to obtain estimates of the driving-force. These estimates are then transformed to the frequency domain by the STFT, and are indeed shown qualitatively to represent weak target echoes. In conclusion, the proposed RSA model exploits the ADFE to produce probabilistic information about the unknown random target embedded in the sea surface.

The remainder of this chapter is organized as follows. In Section 3.2.2, we briefly revisit the texture modeling of sea clutter. In Section 3.2.3, we study a controlled case, and demonstrate the algorithm's tracking performance. In Section 3.2.4, we provide the results of experiments performed on the live recorded radar returns compiled from the Dartmouth database.[3] Section 3.3 concludes the chapter with final remarks.

3.2.2 Texture Modeling of Sea Clutter

The RSA requires an approximate model of the radar search space. In sea conditions, this is a sea clutter model. Haykin and Thomson (1998) suggested that the sea clutter is a cyclostationary process. They supported their claim using the Loeve transform. As an independent study, a perceptually satisfying analytic sea clutter model is presented by Gini and Greco (2001), which reached the same result on the cyclostationary character of the sea clutter. Cyclostationarity of the sea clutter is associated with the quasi-periodic structure of the sea surface. In our work, we will be concerned with the model proposed by Gini and Greco (2001), whose results will be briefly reproduced below. According to the texture model (Gini and Greco, 2001), discrete samples of the complex envelope of the clutter process can be written as the product of two components

$$y_n = \sqrt{\tau_n} x_n. \tag{3.36}$$

[3] Available at http://soma.ece.mcmaster.ca/ipix/dartmouth/datasets.html.

In this formulation, x_n is a complex-valued stationary Gaussian process with zero mean and unit power. It is called the *speckle*, and it represents the local backscattering. τ_n is a non-negative real stochastic process, called the *texture.* It is reported (Gini and Greco, 2001) that the correlation length of the speckle is on the order of tens of milliseconds, while that of the texture is on the order of seconds. Farina *et al.* (1997) studied the time-variation of the texture component by simulations, and concluded that the texture slowly fluctuates around a mean value. Ward *et al.* (1990, sec.2) also described the sea clutter as a modulated Gaussian process and demonstrated that the modulating signal (i.e. the texture) changes in a much slower time scale than the local backscatterers (i.e. the speckle). Our own results also confirm the conclusions reached by Farina *et al.* (1997) and Ward *et al.* (1990), as will be illustrated in Fig.3.11. The amplitude of the fast changing speckle component is modulated by the slowly varying texture component in (3.36) as argued by Field and Haykin (2008); Haykin *et al.* (2002); Ward *et al.* (1990). From a physical perspective, high sea waves (represented by the texture) dominate the movement of the local scatterer (the speckle). The movement of a small target on sea surface will also be dominated by the sea waves. Based on these intuitive ideas, we build the following state-space formalism for a target+clutter scenario, which is a more specific form of the general model in (1.1):

$$\begin{aligned} x_{n+1} &= g\left(x_n\right) + u_n + \omega_{n+1} \\ y_{n+1} &= \sqrt{\tau_n} x_n + v_{n+1}. \end{aligned} \tag{3.37}$$

In (3.37), the population of the local scattering x_n is regressed on its own past by the unknown mapping $g(.)$. The target, u_n, is viewed as an unknown additional local scatterer embedded in the sea surface. We introduce ω_n that accounts for the model uncertainty, whereas v_n denotes the measurement noise. The texture component modulates both x_n and u_n as observed in the second line of (3.37). This formality corresponds to the fact that the dynamics of both the local scatterers and a small target are governed by the same global wave structure–the texture. In Fig.3.9(a), we plot a sample window of the target+clutter returns in the spectral domain. In Fig.3.9(b), pure clutter returns are plotted. This graph addresses the effect of close coupling between a small target and the clutter dynamics.

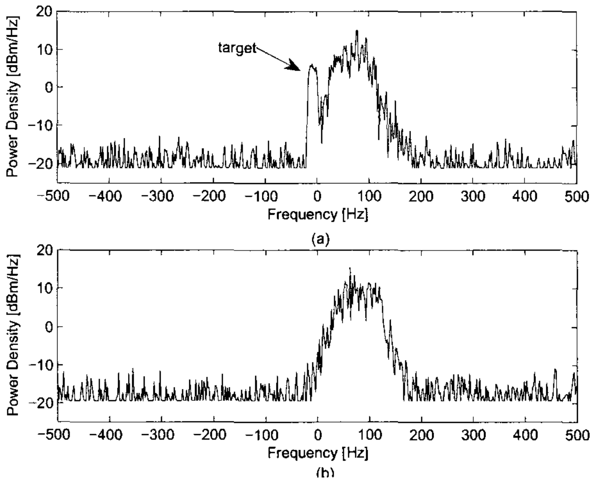


Figure 3.9: (a) Target+clutter, and (b) pure clutter radar returns in Doppler domain. y-axis is spectral power density, and x-axis is Doppler frequency.

Estimation Algorithm

For the application of the ADFE to the estimation of the hidden target signature, let us first reproduce (3.13) obtained in the previous section, in which the prediction error is related to the target signature as given in (3.38),

$$e_{n+1} = \frac{\partial f\left(y_n, u_n\right)}{\partial u_n} u_n + \epsilon_{n+1}. \tag{3.38}$$

Note that $\partial f\left(y_n, u_n\right)/\partial u_n$ is a model-dependent term. Using the sea clutter model in (3.37), we can specify this term as

$$\frac{\partial f\left(y_n, u_n\right)}{\partial u_n} = \frac{\partial y_{n+1}}{\partial u_n} = \sqrt{\tau_n}. \tag{3.39}$$

Incorporating this result in the last line of (3.38), and invoking the noise-smoothing operation as described in (3.14), the output of the noise-smoothing filter can be now described as

$$r_{n+1} = \sqrt{\tau_n} u_n + \epsilon_{n+1}. \tag{3.40}$$

Note that the nature of the radar application under consideration enables us to obtain the estimates of the texture component $\sqrt{\tau_n}$ in (3.40) by means of a separate filter. To elaborate, imagine that the BTT returns the range bins that are labeled not to contain a target with high confidence. Then the texture component can be estimated on a sliding-window basis as suggested by Farina *et al.* (1997) from a nearby target-free range bin. Thus, the estimates of $\sqrt{\tau_n}$ can be incorporated into the regularized adaptive estimator as shown in Fig.3.10, where the output of the estimator is scaled by $1/\sqrt{\hat{\tau}_n}$ at the rightmost block.

To demonstrate how the estimates of the texture are obtained, let us consider that a target-free range is addressed by the BTT. For example, the range bins #4 of file#30 and file#280 are known to be pure clutter (See Table 3.4 for a detailed description of the datasets used in this experiment). Using the data available in those

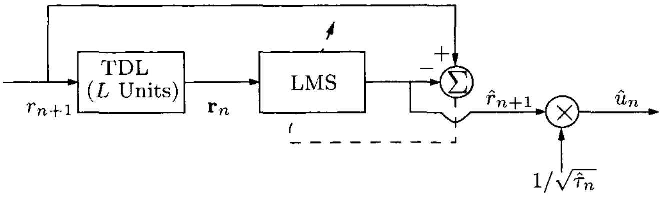


Figure 3.10: Operation of the regularized adaptive estimator in Fig.3.1 is modified with the incorporation of estimates of the texture component, $\sqrt{\tau_n}$ as observed in the rightmost block. The other operating blocks to the left of the "TDL" herein are identical to those in Fig.3.1.

range bins, we estimate $\sqrt{\tau_n}$ using the methodology developed by Farina *et al.* (1997) as given in (3.41),

$$\hat{\tau}_n = \frac{1}{2\psi} \sum_{m=1}^{\psi} |y_n^m|^2, \tag{3.41}$$

where n is the running window index, ψ is the window length, and y_n^m is the mth sample drawn from the nth sliding window of a target-free range bin. Following from (3.41), $\sqrt{\hat{\tau}_n}$ is simulated using real data gathered from the range bins #4 of file#280 and file#30 for varying window length. The slow variation of the texture component is illustrated over a duration of 5 seconds for both datasets. The results are shown in Fig.3.11(a) and Fig.3.11(b) for file#280 and file#30 respectively. It is observed that for $\psi = 256$ and $\psi = 512$, the effect of rapidly varying speckle component is still visible as argued by Farina *et al.* (1997). For $\psi = 1024$, the speckle component is isolated out, yet the variation of $\sqrt{\hat{\tau}_n}$ still follows the general trend. For $\psi = 2048$ however, $\sqrt{\hat{\tau}_n}$ is oversmoothed. These results are in close agreement with those presented in (Farina *et al.*, 1997).

A small random target may exhibit an irregular trajectory, which necessiates a radar receiver to be capable of tracking the statistical variations in the environment. To this end, before presenting the algorithm's performance on live-recorded data, we

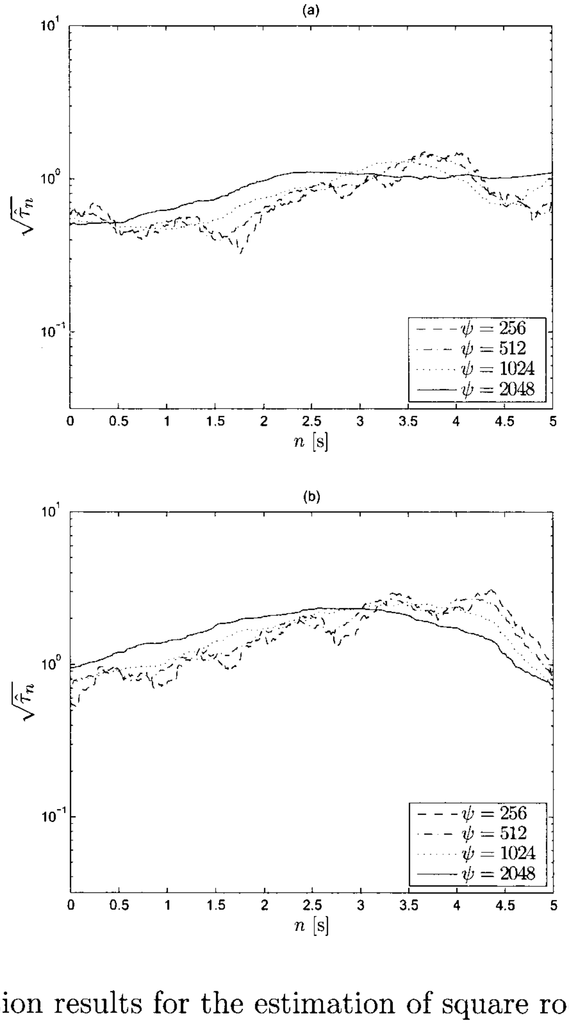


Figure 3.11: Simulation results for the estimation of square root texture component, $\sqrt{\hat{\tau}_n}$ from (3.41) for different window lengths, ψ. (a) file#280, #4. (b) file#30, range bin #4.

first demonstrate the ADFE's tracking capability on a controlled experiment.

3.2.3 Illustrative Example

We study the system given in (3.32) which is perturbed by a driving-force that exhibits a time varying frequency content. Differently from the single-tone driving-force in (3.32), the following setup for the driving-signal is considered herein:

$$u_n = \cos\left(\frac{n}{250}\exp\left(\frac{n}{1000}\right)\right). \tag{3.42}$$

u_n in (3.42) has a time variation in its frequency content, for which the sampling rate is such that $f_s \approx 200 f_{max}$ where f_{max} is the maximum instantaneous frequency in u_n for $n = 2000$.

For simulations, we draw 2000 samples from (3.32) perturbed with the force given in (3.42). Signal-to-process noise power ratio is set to $\frac{P_{u_n}}{\sigma_\omega^2} = -7$ dB to represent a weak target in the state equation. Signal-to-measurement noise power ratio $\frac{P_{x_n/20}}{\sigma_v^2}$ is varied in order to test the algorithm's performance under varying levels of measurement noise.

We conduct some simulations on the ESN bank once again to test the sensitivity of the tracking performance of the ADFE to the selection of the ESN free parameters. In Table 3.3, the time-averaged variance of the estimation error $\sigma^2_{\tilde{u}_n}$ is calculated for some candidate values of the quadruple $\{\varsigma, \rho, N, V\}$. When testing the parameters $\{\varsigma, \rho, N\}$, $V = 10$ is held constant since V has a significant effect on the algorithm's performance. All of the remaining 4^3 combinations of the triple $\{\varsigma, \rho, N\}$ are tried out, and it is observed that different configurations of $\{\varsigma, \rho, N\}$ do not lead to any significant difference in the estimation error. Therefore, when one of the parameters is being tested, the three others are fixed at the following values: $\varsigma = 0.1$, $\rho = 0.1$, $N = 20$, $V = 10$. For instance, in the fourth column of Table 3.3, ρ is under test, and $\varsigma = 0.1$, $N = 20$, $V = 10$ are held constant. As observed from the first column in Table 3.3, using $V \geq 10$ ESNs running in parallel leads to a reduction in the estimation error by a factor of 2. For the selection of $\{\varsigma, \rho, N\}$, the user has the flexibility to assign the values without employing an exhaustive search, which is a desired property.

Table 3.3: candidate values for ESN configuration. $\sigma^2_{\hat{u}_n}$ denotes the time-averaged variance of estimation error.

V	$\sigma^2_{\hat{u}_n}$	N	$\sigma^2_{\hat{u}_n}$	ς	$\sigma^2_{\hat{u}_n}$	ρ	$\sigma^2_{\hat{u}_n}$
1	0.1512	20	0.0724	0.05	0.0769	0.1	0.0674
10	0.0803	30	0.0730	0.1	0.0739	0.4	0.0819
15	0.0695	40	0.0742	0.15	0.0796	0.7	0.0904
50	0.0650	50	0.0786	0.2	0.0730	0.9	0.0876

For a visual illustration of the algorithm's performance for the estimation of u_n in (3.42) from the system given in (3.32), we plug the following parameters based on the procedures explained above, and run the algorithm 100 times: ESN bank: $\varsigma = 0.1$, $\rho = 0.1$, $N = 20$, $V = 50$; decaying constant for the noise-smoother: $\Gamma = 0.9$, the adaptive estimator: $L = 100$, $\lambda = 1$. The average of 100 estimates of $\hat{u}_n$ is plotted vs. the original signal, u_n in Fig.3.12(a). As expected, the estimator keeps track of the original signal after the adaptation period is completed.

Although the averaging mechanism introduced in the ESN bank approach provides a remarkably greater immunity against the measurement noise than a single ESN, the

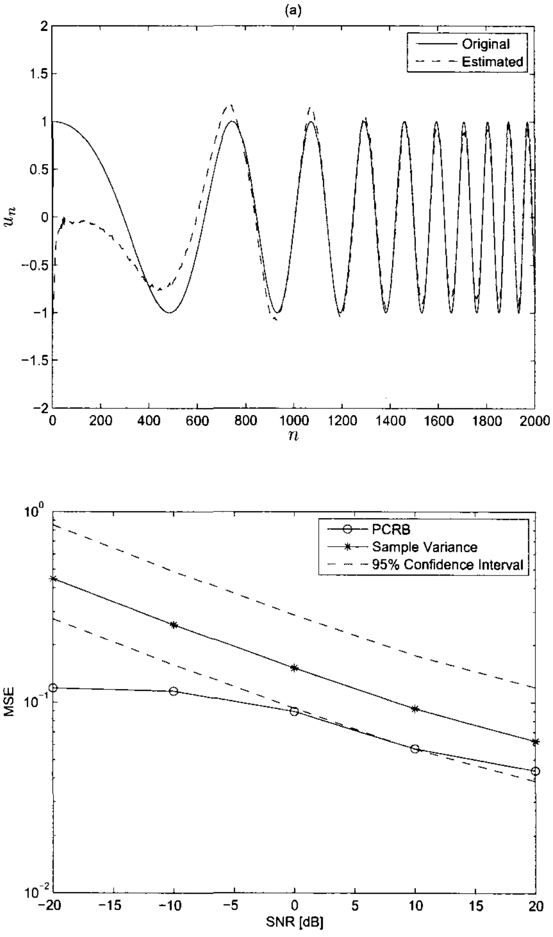


Figure 3.12: Simulation results for the recovery of u_n from (3.32). (a) Solid line is the original driving-force u_n. Dashed curve is the average reconstruction obtained by 100 independent runs of the estimator. (b) Sample variance $\sigma^2_{\hat{u}_n}$ and 95% confidence interval vs. the PCRB for varying SNR.

smoothness of the unknown hybrid mapping $f(.)$ may be violated for high measurement noise levels. To this end, we simulate the time-averaged variance of the estimation error for varying signal-to-measurement noise ratios $(-20, -10, 0, 10, 20$ dB) to obtain a quantitative measure of the algorithm's noise performance. In Fig.3.12(b), we plot the estimation variance, the 95% confidence interval and the PCRB. Despite its looseness, the PCRB can be interpreted as the true variance, and its distance from the confidence interval provides a performance assessment of the estimator. The result is that, the algorithm performs well for moderate to high SNR levels (e.g. 0 to 20 dB), whereas its performance is largely degraded for very noisy sensor models (e.g. -20 dB) exemplified by a large deviation from the PCRB as illustrated in Fig.3.12(b).

3.2.4 Experiments with Live Recorded Data

Dataset Description

In this section, we perform our experiments on the radar returns collected with the McMaster IPIX (Intelligent PIXel Processing) radar at the east coast of Canada, in Dartmouth, Nova Scotia. The radar is operated in the dwell mode with a 1° pencil beam and a fixed radio frequency at 9.39 GHz. It is mounted on a cliff top at 30 m above the sea level. The target is a 1 m diameter styrofoam ball wrapped in radar reflecting material and floats on ocean surface on anchor line at about 2.5 km off shore. Effective pulse repetition frequency is 1 kHz. Each data file in the Dartmouth database contains the I and Q channel measurements for about two minutes. Each data file is divided into 14 range bins, whose ranges are separated by 15 meters. Some of these range bins contain a weak target. We conduct our experiments on the two data files (file #280 and file #30) having the lowest target-to-clutter ratios (TCR).

The characteristics of the range bins of interest are summarized for both data files in Table 3.4.

Table 3.4: Details of file #280 and file #30 from the Dartmouth database

	Range Bin	Range	Azimuth	Elevation	TCR
F.#280	#7	2640 m	179°	359°	−7 dB
	#8	2655 m	179°	359°	0 dB
F.#30	#6	2649 m	128°	359°	−3 dB
	#7	2664 m	128°	359°	0 dB

We are going to perform our experiments on the secondary range bins (bin #7 for file #280 and bin #6 for file #30), both of which are known to contain a weak target. The same targets however show much stronger in the primary range bins which makes it possible to retrieve the target trajectory based on visual inspection. Therefore we will refer to the Time-Doppler (TD) images of the primary range bins (bin #8 for file #280 and bin #7 for file #30) for performance assessment.

As seen in Table 3.4, range, azimuth and elevation dimensions are fixed for each of the particular range bins that are used in our experiments. Then the radar returns can be conveniently visualized by plotting the spectral measurements vs. time and Doppler velocity omitting the other dimensions that are constant. This way, we obtain the three-dimensional TD images as displayed in Fig.3.13 and Fig.3.14. These TD images can be linked with the sensing geometry explained in Section 3.2.1 in the following way: The y-axis in each image represents the resolution cells, which are indicated only by their Doppler dimensions since the other dimensions are fixed. Each pixel in a TD image represents the spectral content of the corresponding resolution cell at time n. Consequently, the time evolution of the target tracks and the clutter returns can be monitored from the TD images in Fig.3.13 and Fig.3.14.

Results

In this section, the hidden target's signature is retrieved from the unknown radar search space using the ADFE given in (3.40), and the sliding-window based estimates of $\sqrt{\tau_n}$ as described in (3.41). We proceed to the experiments by setting $\psi = 1024$, and substituting $\sqrt{\hat{\tau}_n}$ in the predictor in (3.40). We apply the hidden input estimator to I and Q components separately. The Doppler spectra for all experiments is estimated on 64 sample sliding window basis. The ESN bank parameters are set to $N = 20$, $\varsigma = 0.1$, $\rho = 0.1$, and $V = 50$.

The Doppler spectra of the raw radar returns in the range bin #8 of the file#280 are shown in Fig.3.13(a). The algorithm's performance is shown on the range bin #7, file#280 in Fig.3.13(b). Comparing Fig.3.13(a) and Fig.3.13(b), it can be seen that the target tracks are greatly highlighted for most time instants. The algorithm however picks up some clutter effects around $45, 60, 70, 90, 100, 120$ s, when the clutter is very strong in the corresponding resolution cells. The results provided are the average of 10 independent runs of the algorithm.

In Fig.3.14(a), we sketch the time-Doppler image for the raw returns in file#30, range bin #7. Fig.3.14(b) shows the algorithm's performance on the range bin #6. For this dataset, the wind is blowing perpendicular to the radar beam. As a result, the visual separation between the target tracks and the clutter is not as clear as it was for Fig.3.13(a), and it requires a more careful inspection to observe that the target tracks are highlighted in Fig.3.14(b). The evolution of the target tracks and the algorithm's performance can be best inspected by comparing Fig.3.14(a) and Fig.3.14(b) at the

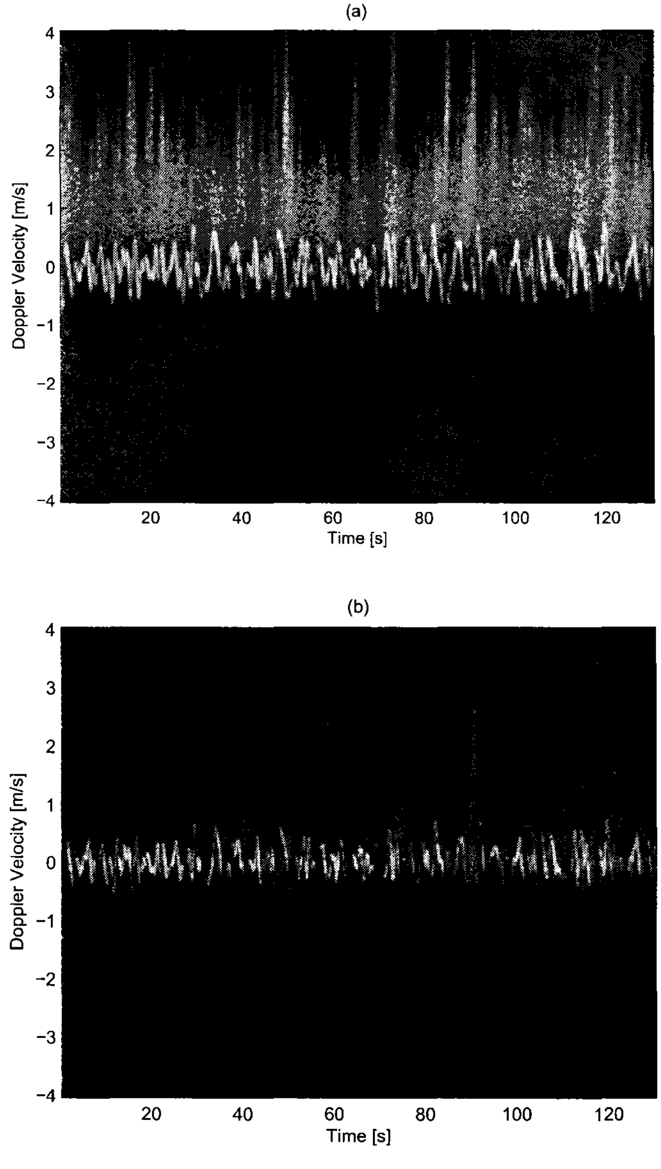


Figure 3.13: Experiments with live-recorded radar returns, file#280-Dartmouth database

(a) Raw radar returns for the file#280, range bin #8, in which the TCR is approximately 0 dB. The bottom trace represents the target Doppler variation with time, hence the desired response. The traces at higher Doppler velocities are the clutter echos. Note that due to the high TCR in this range bin, the target returns are strong and hence bright enough to visually distinguish the desired response from the background clutter echos.

(b) Results of the target signature search algorithm applied to the file#280, range bin #7, in which the TCR is 7 dB lower than the TCR in the upper graph.

time instants $30, 40, 60, 65, 70, 85$ s, and after 100 s, when the target returns are easier to visualize. Also, the effect of strong clutter returns are visible at some time instants in Fig.3.14(b) such as around $50, 60, 100, 110$ s.

3.3 Conclusion

We described an estimation algorithm for the target template search as a realization of the radar scene analysis (RSA) for a cognitive radar receiver. We demonstrated the accuracy of the algorithm in a quantitative manner on a controlled experiment using the posterior Cramer-Rao lower bound. For the real-world radar application, the desired signal is available only visually, hence the algorithm's performance is judged that way. In both Fig.3.13(b) and Fig.3.14(b), we see that for the time instants in which the clutter returns are strong, the algorithm picks up some clutter effects. Refinement of these results is possible by propagating the belief in the target returns in time, which suits the spirit of Bayesian inference for post-processing. We have conjectured the use of the developed method to fit a statistical distribution to the target returns, and infer the target transition probabilities on the fly. Specifically, the RSA can continuously update and provide the following information to the BTT on the basis of the proposed target signature search algorithm:

- probability density function of the target returns $p\left(\mathbf{d}_n|\xi_n^r\right)$;
- probabilistic knowledge on the target agility $\mathbb{P}\left(\xi_{n-1}^q|\xi_n^r\right)$.

These two pieces of information will be exploited by the Bayesian target tracker (BTT), the unit which will make the final decisions about the target tracks. Then the decisions made by the BTT will be fed back to the RSA, and the target search

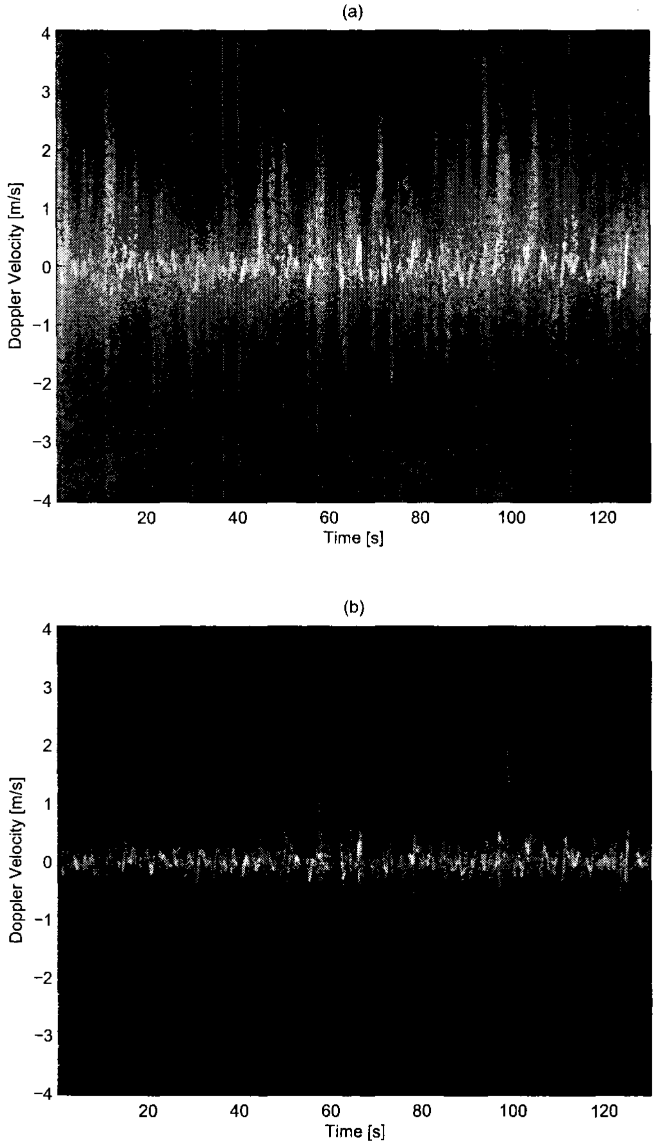


Figure 3.14: Experiments with live-recorded radar returns, file#30-Dartmouth database.

(a) Raw radar returns for the file#30, range bin #7, in which the TCR is approximately 3 dB. The trace around the 0 Doppler velocity represents the target Doppler variation with time, hence the desired response. The traces at higher Doppler velocities are the clutter echos. Note that due to the high TCR in this range bin, the target returns are strong and hence bright enough to visually distinguish the desired response from the background clutter echos.

(b) Results of the target signature search algorithm applied to the file#30, range bin #6, in which the TCR is approximately 6 dB lower than the TCR in the upper graph.

algorithm will be re-initiated. The results of ongoing research efforts on the realization of a Bayesian target tracker supported by the proposed radar scene analysis strategy will be reported in the next article.

Chapter 4

Bayesian, Maximum-Likelihood, and Prediction Error Methods for the Estimation of Hidden Inputs From Nonlinear Dynamic Systems

This chapter covers the material in the article (Güntürkün *et al.*, 2010) that is prepared for submission. Introductory notes (e.g. case examples, literature review) in (Güntürkün *et al.*, 2010) are omitted in this chapter since those notes are already given in greater detail in Chapter 1. The description of the driving-force estimator is also skipped herein since it was analyzed in detail in Chapter 3. The objective of this chapter is to provide deeper insights into the capabilities and limitations of the ADFE. Applications of the Bayesian techniques to the estimation of hidden dynamic inputs are studied to pursue this objective, since the Bayesian methods provide practical merits of performance assessment under varying observability conditions. The Bayesian techniques used in this chapter are built on the background material provided in Chapter 2.

4.1 Introduction

In this chapter, we report the results of recent research efforts on the estimation of hidden inputs from nonlinear dynamic systems. The hidden inputs or the driving-forces refer to the unobservable perturbations embedded in the system dynamics. In two previous chapters, we addressed an adaptive driving-force estimator (ADFE). The ADFE is designed for reconstructing the hidden inputs using only the time series observations without the availability of an analytical description of the system. The resulting algorithm was shown to perform well on a variety of tasks exemplified by some nonlinear stochastic/chaotic systems. In this chapter, we provide a deeper insight into the performance and limitations of the algorithm. Specifically, we interpret the hidden input estimation problem from missing data - maximum likelihood (ML), and Bayesian inference perspectives. As opposed to the ADFE, these approaches require partial or full prior knowledge of the underlying physical phenomena respectively. We provide the performance results of the ADFE, ML and Bayesian techniques on a nonlinear stochastic system where the hidden input is a rapidly changing additive term in the state equation. All the filters are realized at the same level of computational complexity for a fair comparison. Experimental results show that the ADFE is a favorable method except for such cases where an evolutionary model for the hidden input is available as well as the governing system equations.

In most real world examples of the type we are considering, a mathematical model describing the underlying phenomenon is constructed. Then the problem is to estimate the hidden information exploiting the mathematical model of the system. The Bayesian inference approach regularizes solutions of such inverse problems by virtue of incorporating all the prior knowledge on the system in a probabilistically optimum fashion. However, in many other cases, (i.e. voice production, brain activity, sea dynamics, an unknown multipath wireless medium, solar system, evolution of seismic waves, etc.) there may be too little or no prior knowledge to build a specific and accurate model of the underlying system with Bayesian methods. A common need in such cases is to reconstruct the hidden driving-forces using only a finite set of time series data without the availability of an analytic model. It is these cases that we address in this article.

In the current chapter, we present deeper insight on the computational complexity, the performance and the limitations of the ADFE. Specifically, we seek to answer the following questions:

1. How accurately could the driving-forces be estimated if the underlying system model were
   (a) fully available (i.e. the state-space pair and an evolutionary model of the driving-force are all known)?
   (b) or partially available (i.e. only the state-space equations are known)?
2. How good is the performance of the ADFE compared to the more optimistic observability situations above under varying environmental conditions?

The question (1a) is addressed by incorporating all the prior knowledge by virtue of

Bayesian inference, which is realized by the Rao-Blackwellized Particle Filter (RBPF), which has been described in Section 2.2.2. The question (1b) is explored by interpreting the problem from a missing-data maximum likelihood (ML) perspective. To this end, the Expectation Maximization (EM) algorithm is combined with a particle filter (PF), which we abbreviate as the EM-PF algorithm as given in Section 2.2.3. The answer to the question (2) is presented by analyzing the performances of the EM-PF, RBPF and ADFE on a nonlinear stochastic system. All of the EM-PF, RBPF and the ADFE are realized at the same computational complexity for a fair comparison. The computational complexity is specified by the number of floating point operations (*flops*).

Before proceeding to the experiments in Section 4.2, let us mention the practical challenges for the operation of the ADFE. As depicted in Fig.3.1 in Section 3.1.2, the first step in the algorithm is to approximate the observables in an online fashion using a bank of ESNs. An underlying assumption for accomplishing this objective is the differentiability of the hybrid mapping, $f(.)$ given in (3.3). However, observing (3.3), the noise processes can be greatly amplified by the nonlinear functions $g(.)$, $h(.)$, or $h^{-1}(.)$. This may harm the continuity, hence the differentiability of $f(.)$, in which case the predictive modeling of the observables may fail.

Second, consider the term $\partial f(y_n, u_n)/\partial u_n$ in (3.13). As mentioned in the previous chapter, this term is model-dependent, and abrupt variations in $\partial f(y_n, u_n)/\partial u_n$ can cause a substantial loss in the algorithm's performance. For the two chaotic systems in Section 3.1.4, the model dependent terms $\partial f(y_n, u_n)/\partial u_n$ have been shown to be time-varying and being nonlinearly dependent on u_n in (3.28) and (3.29) respectively, thus posing particular challenges for the estimation of u_n. For the stochastic system in (3.32) however, where we had $\partial f(y_n, u_n)/\partial u_n = 1/20$, where the measurement equation was linear in state. Here, we modify (3.32) by making the measurement model nonlinear as given in (4.1),

$$\begin{aligned} x_{n+1} &= 0.5x_n + \frac{25x_n}{1+x_n^2} + u_n + \omega_{n+1} \\ y_{n+1} &= \frac{x_{n+1}^2}{20} + v_{n+1} \end{aligned} \tag{4.1}$$

thus making the estimation of u_n more involved. For the system in (4.1), we have $\partial f(y_n, u_n)/\partial u_n = x_{n+1}/10$ which suggests that the process noise may have a harmful effect on the performance of the estimator. Therefore, analyzing the estimator's performance on (4.1) for varying levels of variance of process noise σ_ω^2, we can gain a deeper understanding of the algorithm's capability in coping with the effect of outliers. As in the previous chapter, the effect of environmental conditions will be quantified with a reference to the PCRB.

4.2 Bayesian Inference and Missing Data - Maximum Likelihood Methods for Hidden Input Estimation

Note that the ADFE is designed for the estimation of driving-forces in such cases where the description of the underlying system is not available at all. Here, we consider the cases where some knowledge regarding the state-space model is available. To elaborate, first let us introduce a dynamic model to describe the evolution of the driving-force u_n, and incorporate this model as an extra state variable into (4.1) to obtain the augmented system description as given in (4.2) in a similar way to the model in (2.21),

$$\begin{aligned}\mathbf{u}_{n+1} &= \mathbf{B}\mathbf{u}_n + \xi_{n+1} \\ x_{n+1} &= 0.5x_n + \frac{25x_n}{1+x_n^2} + \mathbf{b}\mathbf{u}_n + \omega_{n+1} \\ y_{n+1} &= \frac{x_{n+1}^2}{20} + v_{n+1}\end{aligned} \tag{4.2}$$

where the $\mathsf{P} \times \mathsf{P}$ regression matrix $\mathbf{B}$ is similar to (2.20), and $1 \times \mathsf{P}$ vector $\mathbf{b} \triangleq [1, 0, \ldots, 0]$ is again defined in a similar way as in the model in (2.21).

4.2.1 Experimental Setup

The experiments in this chapter are conducted on (4.2), where the objective is to estimate u_n. The estimators are specified for the varying observability conditions in the following manner:

1. Fully observable dynamics: For the application of the RBPF within this case, we construct u_n in (4.1) to be a 2nd-order autoregressive process (i.e. $\mathsf{P} = 2$ in (4.2)). Then the regression coefficients $[c_1, c_2]^T$ in (2.18) are calculated using the covariance method so as to minimize the forward prediction error in the least squares sense.[1] Setting $\mathsf{P} = 2$ is sufficient since it leads to a negligibly small variance of modeling error σ_ξ^2. Using $\mathsf{P} > 2$ does not lead to a significant decrease in σ_ξ^2. This application will be referred to as RBPF-AR(2), for which the model and all parameters are assumed known.

2. Partially observable dynamics:

[1] The command `arcov` is readily available in Matlab for this operation.

- In this case, the first method we propose is to apply the RBPF without the availability of the model coefficients, $[c_1, \ldots, c_{\mathsf{P}}]^T$. Then a natural way to proceed with the RBPF is to use a random walk model (Liu and West, 2001) for the evolution of the driving-force. This filter is abbreviated as the RBPF-RW.
- The second method we propose for this estimation situation is the EM-PF algorithm. The objective of the EM algorithm is to estimate the coefficient vector $[c_1, \ldots, c_{\mathsf{P}}]^T$.

3. Finally, for such cases where there is no prior knowledge on the system dynamics, the proposed ADFE is invoked without the knowledge of (4.1).

4.2.2 Estimation Performance with Minimal Complexity

Our objective in the first part of the experiments is to realize all of the methods mentioned above at minimal computational cost. In Appendix B, the computational complexities of all estimators (i.e. the ADFE, RBPF, and EM-PF) are derived. Below, we summarize the computational cost of each of these methods in terms of flop counts:

Computational cost of the ADFE is obtained as:

$$\mathcal{C}_{ADFE}(\varsigma, V, N, L) = \left[(2\varsigma + 5)N^2 + (2\kappa_1 + \kappa_2 + 12 - \varsigma)\, N + 1\right] V + 11L + 10 + \kappa_1, \tag{4.3}$$

where κ_1 and κ_2 are some constants that represent the number of flops per division and exponential operations respectively, as described in Table B.1 in Appendix B.

The computational cost of the RBPF is given as:

$$\mathcal{C}_{RBPF}(M,\mathsf{P}) = 4\mathsf{P}^3+5\mathsf{P}^2+2\mathsf{P}+(2\mathsf{P}^2+5\mathsf{P}+c_1+c_4+c_5+4)M+c_2+c_3+(\mathsf{P}+1)c_6-1, \tag{4.4}$$

where the constants c_1 to c_6 are as; c_1—evaluation and normalization of importance weight per particle, c_2—resampling for all particles, c_3—Cholesky factorization, c_4—random number generation per particle, c_5—evaluation of the nonlinear mapping, $g(.)$, and c_6—division. The details of (4.4) are defined in Table B.2 in Appendix B.

The Computational cost of the EM-PF is shown to be using the complexity of the RBPF:

$$\begin{aligned}\mathcal{C}_{EMPF}(M,\mathsf{P},K) =& K\Big[10\mathsf{P}^3+13\mathsf{P}^2+(4\mathsf{P}^2+6\mathsf{P}+c_1+c_4+c_5+4)M \\ &+c_2+c_3+(\mathsf{P}+1)c_6+c_7-1\Big],\end{aligned} \tag{4.5}$$

where the constants c_1 to c_6 are the same as in given (4.4). c_7 denotes the inversion of a $\mathsf{F}\times\mathsf{P}$ matrix as given in Table B.3 in Appendix B.

We first determine the minimum complexity of the EM-PF method since it is computationally the most demanding filter among the others. Specifically, using 50 particles and 5 EM iterations leads to a reasonable performance for the EM-PF with a model order of $\mathsf{P}=2$. Increasing the computational resources does not lead to a significant increase in the performance for the EM-PF. With this setting, the EM-PF algorithm is realized at approximately 37.10^3 flops. In Table 4.1, this setup is represented by $\mathcal{C}_{EMPF}(50,2,5)$. Then we realize the other estimators at a similar level of computational cost as presented in Table 4.1, where the theoretical and simulated flop counts[2] are shown to agree on a close degree of approximation for each estimator. The discrepancy between the theoretical and simulated flop counts in Table 4.1 is less than 3% for all estimators. We proceed to the experiments with the settings given in the second column of Table 4.1.

Table 4.1: Theoretical and simulated computational complexity of the driving-force estimation methods. The EM-PF is realized by 50 particles with a model order of $\mathsf{P} = 2$ and 5 EM iterations. The RBPF-AR(2) is realized with 275 particles with $\mathsf{P} = 2$. The RBPF-RW is realized with 300 particles, with $\mathsf{P} = 1$ (i.e. the random walk). The ADFE has 34 ESNs in the ESN bank, each having 30% connected dynamic reservoirs with 10 neurons. The regularized LMS estimator has 100 tap-weights.

Filter	Realization	Flop Count	
		Theoretical	Simulated
EM-PF	$\mathcal{C}_{EMPF}(50, 2, 5)$	37125	37594
RBPF-AR(2)	$\mathcal{C}_{RBPF}(275, 2)$	36983	37812
RBPF-RW	$\mathcal{C}_{RBPF}(300, 1)$	36980	38183
ADFE	$\mathcal{C}_{ADFE}(0.3, 34, 10, 100)$	36410	37337

An example run of the ADFE algorithm with the complexity $\mathcal{C}_{ADFE}(0.3, 34, 10, 100)$ (i.e. $\varsigma = 0.3$, $V = 34$, $N = 20$, $L = 100$) is illustrated in Fig.4.1, which demonstrates the accuracy of the proposed method for the reconstruction of a rapidly varying non-smooth driving-force. For the sake of visuality, only the last 50 samples are shown in Fig.4.1. The decaying constant of the noise smoother is $\Gamma = 0.9$.

As stated in the introduction of the current chapter, we conduct our experiments for varying environmental conditions which are represented by changes in the model uncertainty and sensor noise in the model (4.1). Let P_u denote the power of the driving-force signal. We vary the ratio of P_u to the variance of the dynamic noise in the range $P_u/\sigma_\omega^2 \in \{-10, 10\}$ dB with 5 dB increments in order to test the performance

[2] The flops are simulated using the `lightspeed toolbox` (Minka, 2009), which leads to much more accurate flop estimates than the Matlab 5.3.

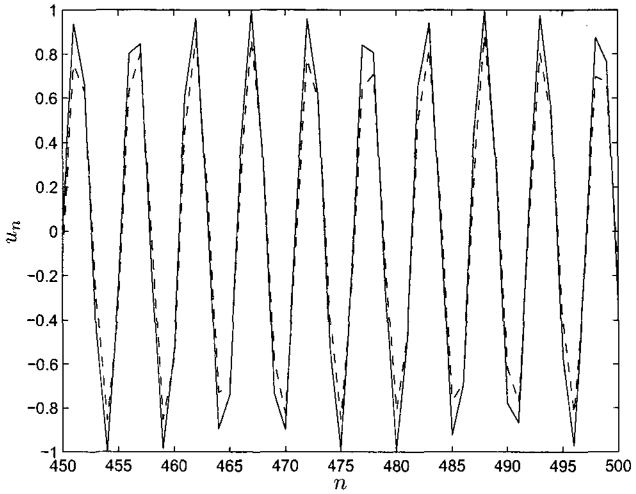


Figure 4.1: An example run of the ADFE. The solid curves denote the actual driving-force, u_n drawn from (4.1). The dashed curves are the estimates provided by the ADFE, which are obtained by a single example run of the ADFE algorithm. The ADFE is realized with the values and the corresponding complexity given in Table 4.1. $\sigma_\omega^2 = 1.6$ (i.e. $P_u/\sigma_\omega^2 = -5$ dB) and $\sigma_v^2 = 0.05$.

of the estimators for varying level of unstructured outliers or model uncertainty. We consider two sensor noise levels. For the first case, we consider a low-noise sensor, and set $\sigma_v^2 = 0.05$. Then we repeat our experiments for the same range of P_u/σ_ω^2 with a high-noise sensor model by setting $\sigma_v^2 = 5$.

The results are shown in Fig.4.2, where the Root Mean-Square Error (RMSE) is plotted with respect to $P_u/\sigma_\omega^2 \in \{-10, 10\}$ dB for all estimators and the Posterior Cramer-Rao Lower Bound (PCRB) for two different measurement noise levels. Comparing the performances of RBPF-RW and EM-PF with the ADFE, it is clear that the ADFE provides a considerably better performance for all noise levels. The RBPF-AR(2) on the other hand greatly outperforms all other filters for all noise levels, follows a similar trend with the PCRB, and gets closest to the PCRB as expected. Inspecting the curves for the ADFE $\sigma_v^2 = 5$ and ADFE $\sigma_v^2 = 0.05$ reveals that the

ADFE is relatively sensitive to both the effect of outliers and the measurement noise for the low-complexity scheme considered in Fig.4.2. Note that the RBPF-RW and EM-PF are simulated only for low measurement noise since both of these filters exhibit a very poor performance (i.e. RMSE> 0.71) for $\sigma_\upsilon^2 = 5$. The estimation errors for all estimators in Fig.4.2 are averaged over 100 independent runs of the respective algorithms. The outputs of the RBPF-RW, EM-PF and the ADFE are normalized by the largest absolute values of their respective estimates so as to compensate the effect of arbitrary scaling. For all particle filtering operations, we use the systematic resampling scheme (Arulampalam *et al.*, 2002; Kitagawa, 1996), due to its ease of implementation and low complexity (Hol *et al.*, 2006).

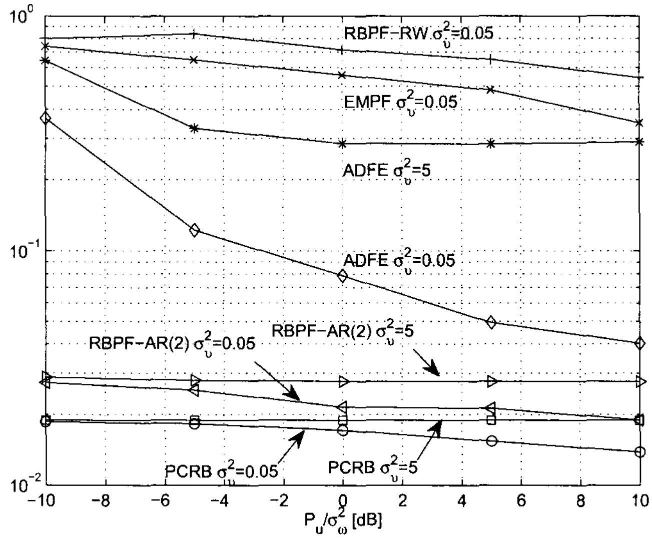


Figure 4.2: Performances of the Rao-Blackwellized Particle Filter, Expectation-Maximization, and the Adaptive Driving-Force Estimator on the estimation of hidden inputs. All filters are realized with a computational cost at approximately $\mathcal{C} = 37.10^3$ flops. The y-axis is the root mean square error (RMSE), while the x-axis is the ratio of the power of the driving-force to the variance of dynamic noise in dB.

4.2.3 The Effect of Computational Resources

Basec on the results presented in Fig.4.2, the following question can be raised: How much better could the ADFE and the other estimators perform if a larger amount of computational resources were available? To address this question, we setup another experiment in which the noise levels are fixed at $\sigma_\omega^2 = 5$ and $\sigma_v^2 = 0.5$. Then we calculate the RMSE for $C \approx \{37, 100, 200, 400\} \times 10^3$ flops for all estimators. The results are presented in Table 4.2. Clearly, for the ADFE, the performance is remarkably improved with more and more computational power becoming available. All the other estimators however perform almost the same accuracy regardless of how much more computational power is available.

Table 4.2: Performance of all estimators vs the increasing computational complexity. The corresponding specifications for the realizations of all filters are provided in Table B.4.

Flop Count	RBPF-RW	EM-PF	ADFE	RBPF-AR(2)
	RMSE			
37.10^3	0.82	0.74	0.62	0.029
100.10^3	0.82	0.73	0.54	0.029
200.10^3	0.83	0.73	0.49	0.029
400.10^3	0.83	0.73	0.43	0.029

Before concluding this subsection, our final set of experiments are designed to specify the conditions under which the ADFE could perform almost at the same level of accuracy as the RBPF. Specifically, we test the performance of the ADFE for $P_u/\sigma_\omega^2 \in \{-10, 10\}$ dB, $C \approx \{37, 100, 200, 400\} \times 10^3$ flops, and compare the results with the RBPF-AR(2).

Note that the RMSE curves in Fig.4.2 were obtained by averaging over 100 runs, since it is feasible to estimate the point variance over a large number of independent trials empirically (e.g. 100) exploiting the LLN. [3] However, for such algorithms that require very large computational complexity, it is more practical to average the results over smaller ensembles (e.g. 10 independent trials) and then refer to some specified tolerance limits for performance assessment. To this end, an interval estimate of the error standard deviation can be obtained in terms of 95% confidence area. This is how we proceed with the ADFE for $\mathcal{C}_{ADFE} \approx \{100, 200, 400\} \times 10^3$ flops. Specifically, in Fig.4.3, we illustrate the point variance estimates for the RBPF-AR(2) with $\mathcal{C}_{RBPF} = 37.10^3$ flops. For $\mathcal{C}_{ADFE} \approx \{100, 200, 400\} \times 10^3$ flops however, we illustrate the 95% confidence intervals (See Appendix B.5). Then a practically meaningful way to judge the performance of the ADFE is to address under what conditions the standard deviation of the RBPF lies within the 95% confidence interval for one of the ADFE realizations. Inspecting Fig.4.3, we observe that only for $P_u/\sigma_\omega^2 \geq 5$ dB and $\mathcal{C}_{ADFE} \geq 400.10^3$ flops the standard deviation of the RBPF lies within the tolerance limits provided by the ADFE.

4.2.4 Discussion

The results of the experiments studied in this section reveals some interesting observations:

1. The essence of the Bayesian inference is to incorporate all the prior knowledge on the underlying phenomena in a probabilistically optimum fashion to solve the inverse problems. Hence, the Bayesian estimators heavily rely on the accuracy of the prior information that is fused in the estimator. As observed in Fig.4.1, the driving-force under study is a dynamic force that exhibits rapid variations

[3] LLN is an abbreviation for the law of large numbers.

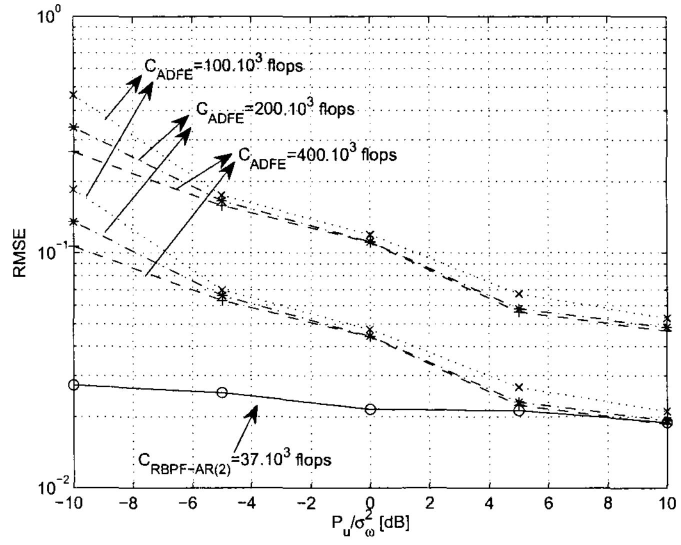


Figure 4.3: 95% Confidence intervals for the standard deviation of estimation error for the ADFE with $\mathcal{C}_{ADFE} \approx \{100, 200, 400\} \times 10^3$ flops, and the point estimate of the standard deviation of the estimation error for the RBPF with $\mathcal{C}_{RBPF} \approx 37.10^3$ flops.

in time. Therefore, a random walk model is a very poor approximation to describe the evolution of the driving-force. This explains why the performance of the RBPF-RW is greatly inferior to all the other estimators under all noise conditions.

The EM algorithm on the other hand performs better than the random walk model for identifying the driving-force evolution. Specifically, consider the lowest noise levels, $(\sigma_\omega^2 = \sigma_v^2 = 0.05)$. In this case, the EM algorithm produces after the 5th iteration an estimate of the coefficient vector in (2.20), $\hat{c}_1 = 0.79$, $\hat{c}_2 = -0.49$. Further iterations do not lead to any remarkable change in the estimate of c_1 and c_2. The actual coefficient vector however is $[c_1, c_2] = [0.72, -1]$. This observation suggests that a slight discrepancy between the actual dynamics and the model on which the Bayesian estimator operates can lead to a substantial decline in the precision of the estimates. Before reaching this claim, we have initialized the EM algorithm with many different initial values of c_1 and c_2. Note that the EM algorithm is very sensitive to the process noise. Therefore, $\sigma_\omega^2 < 0.05$, the results can be further improved, which however is out of the scope of this work.

When the exact prior information is integrated in the solution of the problem, the results are shown to be greatly improved as illustrated by the RBPF-AR(2) curves in Fig.4.2.

As a result, we conclude that unless full prior knowledge about both the governing dynamics of the underlying system and the evolution of the driving-force is available, the proposed ADFE is evidently a favorable approach for the driving-force estimation problem.

2. A shortcoming of the ADFE is its sensitivity to the increasing noise levels in the system. Note that the ADFE was shown to be considerably less sensitive to the process noise when the driving-force is a smooth signal in multiplicative and/or exponent forms (Güntürkün, 2010a). Following from the results presented by Güntürkün (2010a) and those obtained in this work, we can suggest that the ADFE provides close performance to the best available estimator for the additive, non-smooth forces only for low noise levels at the expense of more than 10 times larger computational resources (see Fig.4.3).

4.3 Conclusion

We have provided a deeper analysis of the adaptive driving-force estimator (ADFE) presented by Güntürkün (2010a,b), which is designed for estimating the hidden inputs (driving-forces) from nonlinear system with unknown dynamics. The emphasis of the current article is on addressing the conditions under which the ADFE is a good choice for hidden input estimation problems. By virtue of experiments conducted on a general nonlinear system, we have demonstrated that if the underlying system is not fully specified by a mathematical model that describes both the governing system dynamics (state-space equations) and the evolution of the hidden force, then the ADFE is indeed a favorable method. It is also illustrated by a statistical analysis that if very large computational resources are available, then the accuracy of the ADFE gets closer to the best available fully coherent Bayesian estimator for low noise levels. The ADFE owes its good performance to the selection of two robust recursive nonlinear filters that are capable of adapting to the environment: A bank of Echo State Networks for predictive modeling of the observables from the unknown system; and the regularized Least Mean Squares filter for the refinement of the raw driving-force estimates. In an earlier article (Güntürkün, 2010b), we demonstrated the usefullness of the ADFE on a real world problem with live recorded data. Specifically, the signature of small random targets were extracted by the ADFE to be utilized in a cognitive radar receiver. Two other venues where the ADFE could be a useful technique are the biomedical and chaotic communications applications. For the former application, the ADFE will be considered for the extraction of perturbation inputs or artifacts from biomedical time series (e.g. EEG data), in which case a specific model of the underlying dynamics is really hard to build. For the latter case, we conjecture the exploration of applicability of the ADFE for detecting message symbols that are embedded in a chaotic carrier.

Chapter 5

Concluding Remarks

We have presented the theory and applications of an estimation algorithm in this thesis, whose objective is to reconstruct the hidden forces from nonlinear nonstationary systems with unknown dynamics. The contributions of the approach can be summarized as follows:

- The literature on the estimation of driving-forces or hidden inputs prior to the presentation of the proposed method has been limited to the estimation of very slowly changing or constant perturbations as noted in Chapter 1. The proposed estimator on the other hand is shown to be capable of retrieving both slowly and rapidly varying driving-forces as well as those forces that exhibit nonstationary or irregular time-variations as demonstrated in Fig.3.3(a), Fig.3.4(a), Fig.3.6, Fig.3.12(a) and Fig.4.1.

- Moreover, the driving-force estimator presented herein is compared with another favorable method (Verdes *et al.*, 2006) on the estimation of slowly varying hidden inputs when the time series are corrupted by additive noise. It is shown in

Table 3.2 that the proposed method provides a significantly greater immunity to the additive noise.

- The conditions under which the hidden forces can be estimated using nonlinear predictors are clearly addressed as given in Proposition 1. Specifically, it is shown that the differentiability of the mapping that governs the unknown system and the universal approximation capability of the predictor are required for a successful application of the estimator. Based on these theoretical observations, the algorithm is put to test under varying environmental conditions exemplified by the varying degrees of uncertainty and outliers in the state equation, and varying amounts of sensor noise in the measurement model. In Fig.3.3(b), Fig.3.4(b), Fig.3.5, Fig.3.12(b) and Fig.4.2, the algorithm is shown to perform reasonably well for moderate to low noise levels.

- The validity of the approach is demonstrated on a real world example in Section 3.1.5, for the estimation of sun's magnetic flux from the sunspot time series without the availability of an analytic description of the underlying phenomenon. Taking the results of the analytic studies by Solanki *et al.* (2002) and Lockwood *et al.* (1999) as true references, the driving-force estimator is demonstrated to capture the main characteristics of the underlying force.

- The radar scene analysis problem is addressed using the proposed estimator for a cognitive radar reception system. Specifically, the signature of a small random target is extracted from the raw sea returns with the algorithm based on a generic model of the sea clutter. Accurate prior knowledge on the statistics of an unknown target is required by the target tracking mechanism for the cognitive radar. To this end, it is shown in Fig.3.13 and Fig.3.14 that the target signature provided by the algorithm indeed enables to obtain such statistical information for the unknown targets. To the author's best knowledge, a specific structure for the realization of a radar scene analyzer is addressed for the first time in the literature (Güntürkün, 2010b).

- As stated in the first paragraph, the proposed algorithm does not assume a mathematical model of the unknown system. In this regard, the proposed method can be classified as a blind estimator. Then the following point is of practical interest: Could better algorithms be employed under more optimistic observability conditions (i.e. the system under study is partially or fully identifiable)? This question is addressed by comparing the method with semi-blind and fully coherent estimators, which are built on the Missing Data-Maximum Likelihood and Bayesian Inference methods, and which require partial and full availability of an analytic model respectively. Providing the same amount of computational resources to all algorithms, it is shown that the proposed approach greatly outperforms the semi-blind estimators as illustrated in Fig.4.2. The fully coherent Bayesian estimator on the other hand is shown to provide the closest performance to the theoretical lower limit, which requires the availability of a mathematical model of not only the governing dynamics of the system of interest, but a generating model of the driving-force too.

- A detailed analysis of the computational requirements of the proposed algorithm and the semi-blind and coherent estimators are addressed in Appendix B. Based on these results, it is shown in Table 4.2 that the performance of the algorithm can be improved by increasing the computational resources. Moreover, it is shown by a statistical analysis in Fig.4.3 that if more than 10 times larger computational resources are made available to the proposed approach than to the fully coherent Bayesian estimator, then its performance approaches to that of the coherent estimator.

Based on the theoretical and practical results provided in this thesis, we conjecture to pursue our future research on the following topics:

- An immediate extension of the work presented in this thesis is to merge the proposed radar scene analyzer structure with a Bayesian target tracking unit so as to obtain the first realization of an actual cognitive radar receiver. Specifically, the statistics of the unknown target returns obtained by the proposed method will be supplied to the Bayesian target tracker. We expect the accurate and online availability of the statistical information on the target returns to greatly facilitate the operation of the Bayesian target tracker.

- Another problem that is well-suited for the application of the proposed method is the processing of biomedical time series for artifact extraction. In particular, we consider the EEG (electroencephalogram) signals, which are used for distinguishing epileptic seizures from other types of spells, and for many other purposes in clinical applications. The driving-force for an EEG record can be controlled (e.g. the blink of an eye), however is difficult to extract from the actual measurements. Hence, the proposed method could be specified for such operations.

- We also consider using the method as a blind chaotic communications receiver. In chaotic communication systems, the security is introduced by virtue of using a chaotic map as the carrier, which efficiently masks the message signal so as to block the access of an eavesdropper. A copy of the carrier signal is made available at the receiver for coherent decoding, which is kept hidden from the third parties. The results of our preliminary experiments on this topic however suggest that the message signal can be retrieved up to a certain accuracy using the proposed driving-force estimator without knowing the carrier. In the future, the probability of error curves will be derived, and it will be explored whether an eavesdropper can actually interfere with the communication link in case they have arbitrarily large computational resources.

Appendix A

Appendix for Ch. 3

A.1 Derivation of the Adaptive Estimator

Let us reproduce the cost function for the regularized adaptive estimator:

$$\begin{aligned} J &= \mathbb{E}\left[\alpha_{n+1}^2\right] + \lambda\|\beta_{n+1}\|^2 \\ &= \mathbb{E}\left[(r_{n+1} - \mathbf{a}_n\mathbf{r}_n)^2\right] + \lambda\|\mathbf{a}_n^T\mathbf{r}_n - \mathbf{a}_{n-1}^T\mathbf{r}(n-1))\|^2. \end{aligned} \tag{A.1}$$

A.1.1 Weight Update

Exact gradient for the weight update is

$$\nabla_{\mathbf{a}_n} J = -2\mathbb{E}\left[\alpha_{n+1}\mathbf{r}_n\right] + 2\lambda\left[\hat{r}_{n+1} - \hat{r}_n\right]\mathbf{r}_n. \tag{A.2}$$

Removing the statistical expectation, we obtain the stochastic gradient update rule,

$$\begin{aligned}\mathbf{a}_{n+1} &= \mathbf{a}_n - \frac{1}{2}\mu_n \nabla_{\mathbf{a}_n} J \\ &= \mathbf{a}_n + \mu_n \left[\alpha_{n+1} - \lambda\beta_{n+1}\right] \mathbf{r}_n.\end{aligned} \tag{A.3}$$

A.1.2 Step-Size Update

The derivative of the cost function with respect to the step-size parameter is

$$\begin{aligned}\frac{\partial J}{\partial \mu_n} &= 2\mathbb{E}\left[\alpha_{n+1}\frac{\partial \alpha_{n+1}}{\partial \mu_n}\right] + 2\lambda\beta_{n+1}\frac{\partial \beta_{n+1}}{\partial \mu_n} \\ &= 2\mathbb{E}\left[\alpha_{n+1}\left(\nabla_{\mu_n}\mathbf{a}_n\right)^T\left(\nabla_{\mathbf{a}_n}\alpha_{n+1}\right)\right] + 2\lambda\beta_{n+1}\left(\nabla_{\mu_n}\mathbf{a}_n\right)^T\left(\nabla_{\mathbf{a}_n}\beta_{n+1}\right) \\ &= -2\mathbb{E}\left[\alpha_{n+1}\Psi_n^T\mathbf{r}_n\right] + 2\lambda\beta_{n+1}\Psi_n^T\mathbf{r}_n,\end{aligned} \tag{A.4}$$

where we have defined $\Psi_n \triangleq \nabla_{\mu_n}\mathbf{a}_n$, which will be also derived iteratively next. In light of (A.4), step-size parameter is updated as

$$\begin{aligned}\mu_{n+1} &= \mu_n - \frac{1}{2}\gamma\frac{\partial J}{\partial \mu_n} \\ &= \mu_n + \gamma\left[\alpha_{n+1}\Psi_n^T\mathbf{r}_n - \lambda\beta_{n+1}\Psi_n^T\mathbf{r}_n\right] \\ &= \mu_n + \gamma[\alpha_{n+1} - \lambda\beta_{n+1}]\Psi_n^T\mathbf{r}_n.\end{aligned} \tag{A.5}$$

Iteration of Ψ_n follows from (A.3):

$$\begin{aligned}
\Psi_{n+1} &= \Psi_n + \left[\alpha_{n+1}\mathbf{r}_n + \mu_n \frac{\partial \alpha_{n+1}}{\partial \mu_n}\mathbf{r}_n\right] - \\
&\lambda \left[\beta_{n+1}\mathbf{r}_n + \mu_n \frac{\partial \beta_{n+1}}{\partial \mu_n}\mathbf{r}_n\right] \\
&= \Psi_n + \left[\alpha_{n+1}\mathbf{r}_n + \mu_n \left(\nabla_{\mathbf{a}_n}\alpha_{n+1}\right)^T \mathbf{r}_n \left(\nabla_{\mu_n}\mathbf{a}_n\right)\right] \\
&- \lambda \left[\beta_{n+1}\mathbf{r}_n + \mu_n \left(\nabla_{\mathbf{a}_n}\beta_{n+1}\right)^T \mathbf{r}_n \left(\nabla_{\mu_n}\mathbf{a}_n\right)\right] \\
&= \Psi_n + \left[\alpha_{n+1}\mathbf{r}_n - \mu_n \mathbf{r}_n^T \mathbf{r}_n \Psi_n\right] - \\
&\lambda \left[\beta_{n+1}\mathbf{r}_n + \mu_n \mathbf{r}_n^T \mathbf{r}_n \Psi_n\right] \\
&= \left[\mathbf{I} - (1+\lambda)\,\mu_n \mathbf{r}_n^T \mathbf{r}_n\right] \Psi_n + \left[\alpha_{n+1} - \lambda\beta_{n+1}\right]\mathbf{r}_n.
\end{aligned} \tag{A.6}$$

A.2 PCRB for The Perturbed Chaotic Maps

Before proceeding to the specific calculations, first we express the system in (3.26) in a suitable form for the derivation of PCRB for the generalized Markovian nonlinear systems (Tichavsky *et al.*, 1998). For this purpose, we adopt an augmented state equation as given in (3.22) on which the PCRB is derived in Section 3.1.3. Here, we specify the general augmented system in (3.22) by fitting an all-pole model responsible for the generation of the dynamic input, u_n. The initial state is also included as a fixed parameter in the augmented state model as given in the first line of (A.7). Let us define $x_n^{(1)} \triangleq x_0$ as the initial state, and $\mathbf{x}_n^{(2)} \triangleq u_n$ as the driving-force in the original system in (3.26). Then substituting $\mathsf{P}_1 = 0$ and $\mathsf{P}_2 = \mathsf{P}$, and $g_1 = 1$, $g_2 = \mathbf{B}$

in (3.22), the augmented system in (3.22) becomes equivalent to

$$\begin{aligned} x_{n+1}^{(1)} &= x_n^{(1)} \\ \mathbf{x}_{n+1}^{(2)} &= \mathbf{B}\mathbf{x}_n^{(2)} + \omega_n \\ y_n &= h\left(x_n^{(1)}, \mathbf{x}_n^{(2)}\right) + v_n, \end{aligned} \tag{A.7}$$

where we have defined $\mathbf{x}_n^{(2)} \triangleq [u_n, u_{n-1}, \ldots, u_{n-\mathsf{P}+1}]^T$, and the first row of the matrix $\mathbf{B}$ holds the autoregressive (AR) coefficients $\{c_\mathsf{P}\}$ for the evolution of driving-force, such that

$$\mathbf{B} = \begin{bmatrix} c_1 & c_2 & \cdots & c_{\mathsf{P}-1} & c_\mathsf{P} \\ 1 & 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{A.8}$$

The process noise vector is defined as $\omega_n = \left[\omega_n^{(2)}, 0, \ldots, 0\right]^T$. Both $\omega_n^{(2)}$ and v_n are zero-mean, white Gaussian processes. The AR coefficients and the variance $\sigma^2_{\omega^{(2)}}$ are determined by solving the Yule-Walker equations using Levinson-Durbin algorithm[1]. The initial state x_0 is uniformly distributed in $[0, 1]$. The measurement model is expressed as a function of the initial state x_0 and the driving-force, u_n mapped by $h : \mathbb{R}^2 \rightarrow \mathbb{R}$.

Determination of the log densities given in (3.25) are essential for the calculation of the posterior information submatrices in (3.24), hence the PCRB.

1. $p\left(\mathbf{x}_{n+1}^{(1)} | \mathbf{x}_n\right)$ is a uniform distribution in the unit interval for the initial state,

[1] The command *aryule* in MATLAB is readily available for this operation.

x_0. Therefore, we have

$$-\log p\left(\mathbf{x}_{n+1}^{(1)}|\mathbf{x}_n\right) = 0; \quad 0 \leq x_0 \leq 1. \tag{A.9}$$

2. $p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right)$ is determined from the second line of (A.7) and its natural logarithm is given by,

$$-\log p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right) = c_1 + \frac{1}{2}\left[\mathbf{x}_{n+1}^{(2)} - \mathbf{B}\mathbf{x}_n^{(2)}\right]^T \Sigma^{-1}\left[\mathbf{x}_{n+1}^{(2)} - \mathbf{B}\mathbf{x}_n^{(2)}\right], \tag{A.10}$$

where c_1 is a constant. In order to avoid the singularity of the covariance matrix Σ, we add little random elements on its main diagonal. Calculations of the gradients of $-\log p\left(\mathbf{x}_{n+1}^{(2)}|\mathbf{x}_n\right)$ w.r.t. $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$ follow from above in a straightforward manner, hence are skipped here.

3. $p\left(y_{n+1}|\mathbf{x}_{n+1}\right)$ is determined from the third line of (A.7) and obtained as,

$$-\log p\left(y_{n+1}|\mathbf{x}_{n+1}\right) = c_2 + \frac{\left(y_{n+1} - h\left(x_{n+1}^{(1)}, x_{n+1}^{(2)}\right)\right)^2}{2\sigma_v^2}. \tag{A.11}$$

Calculation of the derivative of $-\log p\left(y_{n+1}|\mathbf{x}_{n+1}\right)$ with respect to $x_{n+1}^{(1)}$ is a nontrivial task. We can invoke the chain rule to find $\dfrac{\partial h\left(x_{n+1}^{(1)}, x_{n+1}^{(2)}\right)}{\partial x_{n+1}^{(1)}}$, or we can use the Lyapunov exponent approximation,

$$\frac{\partial h\left(x_{n+1}^{(1)}, x_{n+1}^{(2)}\right)}{\partial x_{n+1}^{(1)}} \approx \exp\left(2(n+1)\Gamma\right), \tag{A.12}$$

which follows from the method developed in (Berliner, 1991). It is noticeable that in either case, the same results are obtained.

Calculation of the derivative of $-\log p\left(y_{n+1}|\mathbf{x}_{n+1}\right)$ with respect to $x_{n+1}^{(2)}$ is straightforward. For the logistic map, it follows from the first line of (3.26) that

$$\frac{\partial h\left(x_{n+1}^{(1)}, x_{n+1}^{(2)}\right)}{\partial x_{n+1}^{(2)}} = x_n\left(1 - x_n\right). \tag{A.13}$$

For the Moran-Ricker map, we obtain from the first line of (3.27)

$$\frac{\partial h\left(x_{n+1}^{(1)}, x_{n+1}^{(2)}\right)}{\partial x_{n+1}^{(2)}} = \kappa\frac{x_n^2}{u_n^2}\exp\left[\kappa\left(1 - x_n/u_n\right)\right]. \tag{A.14}$$

Appendix B

Computational Complexity Derivations

B.1 Computational Complexity of the ADFE

Derivation of the computational complexity of the ADFE algorithm is given as

$$\mathcal{C}_{ADFE}(\varsigma, V, N, L) = \left[(2\varsigma + 5)N^2 + (2\kappa_1 + \kappa_2 + 12 - \varsigma)\, N + 1\right] V + 11L + 10 + \kappa_1, \tag{B.15}$$

whose detailed derivation is given in Table B.1 in terms of flop counts.

B.2 Computational Cost of the RBPF

The complexity of the RBPF algorithm is studied by Karlsson *et al.* (2005) and illustrated to be $\mathcal{O}(\mathsf{P}^3)$. For the sake of generality, Karlsson *et al.* (2005) analyzed such cases where both parts of the state model are linear, and the nonlinearity is confined

Table B.1: Computational cost of the adaptive driving-force estimator in terms of floating point operations. V denotes the number of ESNs in the ESN bank, N denotes the number of neurons in each ESN in the ESN bank, c is the connectivity rate of each dynamic reservoir, L denotes the number of taps of the regularized adaptive estimator. κ_1 and κ_2 indicate the number of flops per division and exponential operations respectively.

Instruction	mul.	add.	other
ESN-DR - (hidden layer):			
$\Delta_1 = \mathbf{W}^{(i)}\mathbf{s}_n^{(i)}$	ςN^2	$\varsigma N(N-1)$	
$\Delta_2 = \Delta_1 + \mathbf{w}^{in,(i)} y_n + \phi_{n+1}$	N	$2N$	
$\mathbf{s}_{n+1}^{(i)} = \tanh(\Delta_2) = \frac{\exp(2\Delta_2)-1}{\exp(2\Delta_2)+1}$	N	$2N$	$N(\kappa_1+\kappa_2)$
RLS - output layer:			
$\Delta_3 = \lambda_{RLS}^{-1}\mathbf{s}_{n+1}^{(i)}$	N		
$\Delta_4 = \Delta_3^T \mathbf{P}_n^{(i)}$	N^2	$N(N-1)$	
$\Delta_5 = 1 + \Delta_4 \mathbf{s}_{n+1}^{(i)}$	N	N	
$\mathbf{k}_{n+1}^{(i)} = (\Delta_4/\Delta_5)^T$			$N\kappa_1$
$z_{n+1} = \left(\mathbf{w}_n^{out,(i)}\right)^T \mathbf{s}_{n+1}^{(i)}$	N	$(N-1)$	
$e_{n+1}^{(i)} = y_{n+1} - z_{n+1}^{(i)}$		1	
$\mathbf{w}_{n+1}^{out,(i)} = \mathbf{w}_n^{out,(i)} + \mathbf{k}_{n+1}^{(i)} e_{n+1}^{(i)}$	N	N	
$\mathbf{P}_{n+1}^{(i)} = \lambda_{RLS}^{-1}\mathbf{P}_n^{(i)} - \mathbf{k}_{n+1}^{(i)}\Delta_4$	$2N^2$	N^2	
Subtotal for 1 ESN: $\mathcal{C}_{ESN} =$	$(2\varsigma+5)N^2 + (2\kappa_1+\kappa_2+12-\varsigma)N$		
$e_{n+1} = (1/V)\sum_{i=1}^{V} e_{n+1}^{(i)}$		$V-1$	κ_1
Noise smoother:			
$r_{n+1} = (1-\Gamma)\, e_{n+1} + \Gamma r_n$	2	2	
Regularized LMS predictor:			
$\hat{r}_{n+1} = (\mathbf{a}_n)^T \mathbf{r}_n$	L	$L-1$	
$\alpha_{n+1} = r_{n+1} - \hat{r}_{n+1}$		1	
$\beta_{n+1} = \hat{r}_{n+1} - \hat{r}_n$		1	
$\Delta_6 = \alpha_{n+1} - \lambda\beta_{n+1}$	1	1	
$\Delta_7 = \Delta_6 \mathbf{r}_n$	L		
$\mathbf{a}_{n+1} = \mathbf{a}_n + \mu_n \Delta_7$	L	L	
$\Delta_8 = \psi_n^T \mathbf{r}_n$	L	$L-1$	
$\mu_{n+1} = \mu_n + \gamma\Delta_6\Delta_8$	2	1	
$\Delta_9 = \left[(1+\lambda)\,\mu_n\right]\mathbf{r}_n$	$L+1$	1	
$\psi_{n+1} = \psi_n - \Delta_9\Delta_8^T + \Delta_7$	L	$2L$	
Subtotal for the Adapt. Est. $\mathcal{C}_{AE} =$	$11(L+1)$		
TOTAL	$V\mathcal{C}_{ESN} + \mathcal{C}_{AE} + V + \kappa_1 - 1$		

to the measurement model only. Building on the ideas developed by Karlsson *et al.* (2005), we have studied the complexity of the RBPF for the specific model in (2.21). Once again, we made use of the neutral flop count estimates for the operations such as exp, square root, log, and random number generation. The operations other than multiplication and addition in Table B.2 are the following: c_1: Evaluation and normalization of importance weight per particle. c_2: Resampling for all particles. Since the nonlinear state is 1-dimensional, the Cholesky factorization is reduced to the calculation of a square root. So, $c_3 = sqrt$. c_4: Random number generation per particle, c_5: Evaluation of the nonlinear mapping, $g(.)$. c_6 denotes division. Then the total complexity of the RBPF given in Algorithm 1 becomes

$$C_{RBPF}(M, \mathsf{P}) = 4\mathsf{P}^3 + 5\mathsf{P}^2 + 2\mathsf{P} + (2\mathsf{P}^2 + 5\mathsf{P} + c_1 + c_4 + c_5 + 4)M + c_2 + c_3 + (\mathsf{P} + 1)c_6 - 1. \tag{B.16}$$

B.3 Computational Cost of the EM-PF

The computational cost required for the implementation of the EM-PF algorithm is derived in Table B.3. Note that the operations in the M-Step are repeated after each EM iteration rather than each time iteration. Thus, to calculate the flops per time iteration, these operations must be divided by the length of the dataset. This brings us that if the dataset is long enough (e.g. $T \geq 200$) then the terms divided by T will vanish, and can be neglected. Hence, the computational cost of the EM-PF algorithm

Table B.2: Computational cost of the Rao-Blackwellized Particle Filter. c_1: Evaluation and normalization of importance weight per particle. c_2: Resampling for all particles. Since the nonlinear state is 1-dimensional, the Cholesky factorization is reduced to the calculation of a square root, $c_3 = sqrt$. c_4: Random number generation per particle, c_5: Evaluation of the nonlinear mapping, $g(.)$. c_6 denotes division.

Instruction	mul.	add.	other
Importance weights			$c_1 M$
Resampling			c_2
State pred. & cov.:			
$\hat{\mathbf{u}}^{(i)}_{n+1\|n} = \mathbf{B}\hat{\mathbf{u}}^{(i)}_{n\|n}$	$P^2 M$	$P(P-1)M$	
$\mathbf{C}_{n+1\|n} = \mathbf{B}\mathbf{C}_{n\|n}\mathbf{B}^T + \mathbf{R}$	$2P^3$	$P^2(2P-1)$	
Particle Transition:			
$\Delta_1 = \mathbf{b}\mathbf{C}_{n\|n}$	P^2	$P(P-1)$	
$\Delta_2 = \Delta_1 \mathbf{b}^T + \sigma^2_\omega$	P	P	
$\Delta_3 = chol(\Delta_2)$			c_3
$\Delta_4 = randn(1, M)$			$c_4 M$
$\omega = \Delta_3 \Delta_4$	M		
$\Delta_5 = g\left(x^{(i)}_n\right)$			$c_5 M$
$\Delta_6 = \mathbf{b}\hat{\mathbf{u}}^{(i)}_{n\|n}$	PM	$(P-1)M$	
$\hat{x}^n_{n+1\|n} = \Delta_5 + \Delta_6 + \omega$		$2M$	
Measur. pred.			
$m^{(i)}_{n+1} = x^{(i)}_{n+1} - g\left(x^{(i)}_n\right)$		M	
Innov. cov. (Ω_{n+1})			
$\Delta_7 = \mathbf{C}_{n+1\|n}\mathbf{b}^T$	P^2	$P(P-1)$	
$\Omega_{n+1} = \mathbf{b}\Delta_7 + \sigma^2_\omega$	P	P	
Kalman gain (Υ_{n+1})			
$\Upsilon_{n+1} = \Delta_7 / \Omega_{n+1}$			$P c_6$
Upd. state & cov.:			
$\Delta_8 = m^{(i)}_{n+1} - \mathbf{b}\hat{\mathbf{u}}^{(i)}_{n+1\|n}$	PM	PM	
$\hat{\mathbf{u}}^{(i)}_{n+1\|n+1} = \hat{\mathbf{u}}^{(i)}_{n+1\|n} + \Upsilon_{n+1}\Delta_8$	PM	PM	
$\Delta_9 = \Delta_7 \Upsilon^T_{n+1}$	P^2		
$\mathbf{C}_{n+1\|n+1} = \mathbf{C}_{n+1\|n} - \Delta_9$		P^2	
$\bar{\mathbf{u}}_{n\|n} = (1/M)\sum_{i=1}^{M} \hat{\mathbf{u}}^{(i)}_{n\|n}$		$M - 1$	c_6
Total:	$2P^3 + 3P^2 + 2P + M($ $P^2 + 3P + 1)$	$2P^3 + 2P^2$ $+M(P^2 + 2P$ $+3) - 1$	$M(c_1 + c_4$ $+c_5)$ $+c_2 + c_3$ $+(P+1)c_6$

can be given by

$$\begin{aligned} \mathcal{C}_{EMPF}(M,\mathsf{P},K) =& K\Big[10\mathsf{P}^3+13\mathsf{P}^2+(4\mathsf{P}^2+6\mathsf{P}+c_1+c_4+c_5+4)M+ \\ & c_2+c_3+(\mathsf{P}+1)c_6+c_7-1\Big]. \end{aligned} \tag{B.17}$$

B.4 Higher Complexity for All Estimators

Table B.4 indicates the realizations of all estimators studied in Section 4.2.3 with varying complexities.

B.5 Confidence Intervals for Parameter Estimation

To determine the 95% confidence limits, let $\mathcal{E}$ denote the number of independent trials. Let Ξ denote the sample variance obtained by averaging over $\mathcal{E}$ independent trials. Let us also denote the actual variance that we are looking for by σ^2. The random variable $\mathcal{E}\Xi/\sigma^2$ is known to have a $\mathcal{X}^2(\mathcal{E})$ density (Papoulis and Pillai, 2002). Then the 95% confidence interval for σ^2 is given by

$$\frac{\mathcal{E}\Xi}{\mathcal{X}^2_{0.975}(\mathcal{E})} < \sigma^2 < \frac{\mathcal{E}\Xi}{\mathcal{X}^2_{0.025}(\mathcal{E})}. \tag{B.18}$$

Table B.3: Computational cost of the Expectation Maximization Particle Filter. P denotes the dimensionality of the linear state, c_6 is division, and c_7 is the inversion of a $\mathsf{P} \times \mathsf{P}$ matrix. T denotes the number of samples, and K denotes the number of EM iterations. The vector $\mathbf{c}$ denotes the regression parameters as defined in (2.18)

Instruction	mul.	add.	other
PF & Forward Recursions	$\mathcal{C}_{RBPF}$		
Backward Recursions & E-Step:			
$\mathbf{C}_{n+1\mid n}^{-1}$			c_7
$\mathbf{P}_t = \mathbf{C}_{n\mid n}\left(\mathbf{B}^{[k]}\right)^T \mathbf{C}_{n+1\mid n}^{-1}$	$2\mathsf{P}^3$	$2\mathsf{P}(\mathsf{P}^2-1)$	
$\Delta_1 = \left(\hat{\mathbf{u}}_{n+1\mid\mathcal{T}}^{(i)} - \hat{\mathbf{u}}_{n+1\mid n}^{(i)}\right)$		$\mathsf{P}M$	
$\hat{\mathbf{u}}_{n\mid\mathcal{T}}^{(i)} = \hat{\mathbf{u}}_{n\mid n}^{(i)} + \mathbf{P}_t\Delta_1$	P^2M	P^2M	
$\Delta_2 = \left(\mathbf{C}_{n+1\mid\mathcal{T}} - \mathbf{C}_{n+1\mid n}\right)$		P^2	
$\mathbf{C}_{n\mid\mathcal{T}} = \mathbf{C}_{n\mid n} + \mathbf{P}_t\Delta_2$	P^3	$\mathsf{P}(\mathsf{P}^2-1)$	
$\mathbb{E}\left[\hat{\mathbf{u}}_n\left(\hat{\mathbf{u}}_n\right)^T\right] = \mathbf{C}_{n\mid\mathcal{T}} + \hat{\mathbf{u}}_{n\mid\mathcal{T}}\left(\hat{\mathbf{u}}_{n\mid\mathcal{T}}\right)^T$	P^2	P^2	
$\mathbb{E}\left[\hat{\mathbf{u}}_{n-1}\left(\hat{\mathbf{u}}_{n-1}\right)^T\right] = \mathbf{C}_{t-1\mid\mathcal{T}} + \hat{\mathbf{u}}_{t-1\mid\mathcal{T}}\left(\hat{\mathbf{u}}_{t-1\mid\mathcal{T}}\right)^T$	P^2	P^2	
$\mathbb{E}\left[\hat{\mathbf{u}}_n\left(\hat{\mathbf{u}}_{n-1}\right)^T\right] = \mathbf{P}_{n-1}\mathbf{C}_{n\mid\mathcal{T}} + \hat{\mathbf{u}}_{n\mid\mathcal{T}}\left(\hat{\mathbf{u}}_{t-1\mid\mathcal{T}}\right)^T$	P^2	P^2	
M-Step:			
$\Delta_3 = \sum_{n=1}^{\mathcal{T}} \mathbb{E}\left[\hat{\mathbf{u}}_{n-1}\left(\hat{\mathbf{u}}_{n-1}\right)^T \mid_{\mathbf{m},\Theta^{[k]}}\right]$		P^2	
$\Delta_4 = \sum_{n=1}^{\mathcal{T}} \mathbb{E}\left[u_n\left(\hat{\mathbf{u}}_{n-1}\right)^T \mid_{\mathbf{m},\Theta^{[k]}}\right]$		$(\mathsf{P}-1)$	
$\left(\mathbf{c}^{[k+1]}\right)^T = \Delta_4\Delta_3^{-1}$	$\mathsf{P}^2/\mathcal{T}$	$\mathsf{P}(\mathsf{P}-1)/\mathcal{T}$	$\frac{c_7}{\mathcal{T}}$
$\Delta_5 = \sum_{n=1}^{\mathcal{T}} \mathbb{E}\left[\left(u_n\right)^2\right]$		1	
$\Delta_6 = 2\left(\mathbf{c}^{[k]}\right)^T \Delta_4^T$	$\frac{\mathsf{P}+1}{\mathcal{T}}$	$\frac{\mathsf{P}-1}{\mathcal{T}}$	
$\Delta_7 = \left(\mathbf{c}^{[k]}\right)^T \Delta_3\mathbf{c}^{[k]}$	$2\mathsf{P}/\mathcal{T}$	$(\mathsf{P}-1)/\mathcal{T}$	
$\sigma_\xi^{2,[k+1]} = \left(\Delta_5 - \Delta_6 + \Delta_7\right)/\mathcal{T}$		$2/\mathcal{T}$	$\frac{c_6}{\mathcal{T}}$
Subtotal for the b.r. & EM-Steps: $\mathcal{C}_{EM}(M,\mathsf{P},\mathcal{T}) =$	$3\mathsf{P}^3 + \mathsf{P}^2(M+3+\frac{1}{\mathcal{T}}) + \frac{3\mathsf{P}+1}{\mathcal{T}}$ $+3\mathsf{P}^3 + \mathsf{P}^2(M+5+\frac{1}{\mathcal{T}})$ $+\mathsf{P}\left(M-2+\frac{1}{\mathcal{T}}\right)$ $\frac{+c_6+c_7}{\mathcal{T}} + c_7$		
Total complexity of EM-PF:	$K\left(\mathcal{C}_{EM}(M,\mathsf{P},\mathcal{T}) + \mathcal{C}_{RBPF}\right)$		

Table B.4: Configurations of the Free Parameters of All Estimators for Varying Levels of Ccmplexity. The following values are fixed for the respective estimators: $\mathsf{P} = 1$ for RBPF-RW; $\mathsf{P} = 2$, $K = 5$ for EM-PF; $\mathsf{P} = 2$ for RBPF-AR(2), and $\varsigma = 0.3, N = 10, L = 100$ for the ADFE. M denotes the number of particles, and V is the number of ESNs in the ESN bank for the ADFE.

Flop Count ($\times 10^3$)	RBPF-RW	EM-PF	ADFE	RBPF-AR(2)
	M	M	V	M
37	300	50	34	275
100	825	140	96	750
200	1635	275	192	1500
400	3255	555	386	3000

Bibliography

Andrieu, C. and Doucet, A. (2000). Particle filtering for partially observed gaussian state space models. *J. R. Statist. Soc. B*, **64**, 827–836.

Andrieu, C. and Doucet, A. (2003). Online expectation-maximization type algorithms for parameter estimation in general state space models. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, **6**, 69–72.

Arcak, M. and Kokotovic, P. (2001). Nonlinear observers: a circle criterion design and robustness analysis. *Automatica*, **37**, 1923–1930.

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Proceedings on Signal Processing*, **50**(2), 174–188.

Bakker, R., Rizueno, G. L., and Haykin, S. (2002). Bayesian approach to direct filtering of radar targets in clutter. Technical Report 0, McMaster University, 1280 Main St. W. Hamilton Ontario, Canada, L8S4L7.

Berliner, L. (1991). Likelihood and bayesian prediction of chaotic systems. *Journal of the American Statistical Association*, **86**, 938–952.

Bhattacharyya, S. (1978). Observer design for linear systems with unknown inputs. *IEEE Trans. Automat. Contr.*, **23**, 483–484.

Boutayeb, M., Darouach, M., and Rafaralahy, H. (2002). Generalized state-space observers for chaotic synchronization and secure communication. *IEEE Trans. Circuits Syst.*, **49**, 345–349.

Bruno, M. and Moura, J. (2001a). Integration of bayes detection and target tracking in real clutter image sequences. In *Proc. IEEE Radar Conference*, pages 234–238, Atlanta, GA, USA.

Bruno, M. and Moura, J. (2001b). Multiframe detector/tracker: Optimal performance. *IEEE Trans. Aerosp. Electron. Syst.*, **37**, 925–944.

Casdagli, M. (1997). Recurrence plots revisited. *Physica D*, **108**, 12–44.

Cover, T. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Electron. Comput.*, **14**, 326–334.

Darouach, M., Zasadzinski, M., and Xu, S. (1994). Full-order observers for linear systems with unknown inputs. *IEEE Trans. Automat. Contr.*, **39**, 606–609.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1–38.

DGPS (2010). Division of geological and planetary sciences, california institute of technology; explanation of seismic waves. `http://www.gps.caltech.edu/`.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo methods for bayesian filtering. *Statistics and Computing*, **10**, 197–208.

Eckmann, J., Kamphorst, S., and Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhys. Lett.*, **4**, 973–977.

Fairman, F., Mahil, S., and Luk, L. (1984). Disturbance decoupled observer design via singular value decomposition. *IEEE Trans. Automat. Contr.*, **29**, 84–86.

Farina, A., Gini, F., Greco, M., and Verrazzani, L. (1997). High resolution sea clutter data: statistical analysis of recorded live data. *IEE Proc.-Radar, Sonar Navig.*, **144**, 121–130.

Field, T. and Haykin, S. (2008). Nonlinear dynamics of sea clutter. *International Journal of Navigation and Observation*, **2008**, 860–881.

Gillijns, S. and Moor, B. D. (2006). Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*, **43**, 111–116.

Gini, F. and Greco, M. (2001). Texture modeling and validation using recorded high resolution sea clutter data. In *Proc. IEEE Radar Conference*, pages 387–392, Atlanta, GA, USA.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc.-Radar, Sonar Navig.*, **140**, 107–113.

Guan, Y. and Saif, M. (1991). A novel approach to the design of unknown input observers. *IEEE Trans. Automat. Contr.*, **36**, 632–635.

Güntürkün, U. (2010a). Sequential reconstruction of driving-forces from nonlinear nonstationary dynamics. *Physica D: Nonlinear Phenomena*, **239**(13), 1095 – 1107. issn: 0167-2789, DOI: 10.1016/j.physd.2010.02.014.

Güntürkün, U. (2010b). Toward the development of radar scene analyzer for cognitive radar. *IEEE J. Oceanic Eng., Special Issue on Non-Rayleigh Reverberation and Clutter*, **35**(2), 303 – 313. issn: 0364-9059, DOI: 10.1109/JOE.2010.2043378.

Güntürkün, U., Reilly, J. P., deBruin, H., and Kirubarajan, T. (2010). Bayesian, maximum-likelihood, and prediction error methods for the estimation of hidden inputs from nonlinear dynamic systems. *Signal Processing*. Prepared for submission.

Ha, Q. and Trinh, H. (2004). State and input simultaneous estimation for a class of nonlinear systems. *Automatica*, **40**, 1779–1785.

Haykin, S. (2006). Cognitive radar: A way of the future. *IEEE Signal Processing Mag.*, **23**, 30–40.

Haykin, S. and Thomson, D. (1998). Signal detection in a nonstationary environment reformulated as an adaptive pattern classification problem. *Proc. IEEE*, **86**, 2325–2344.

Haykin, S., Bakker, R., and Currie, B. (2002). Uncovering nonlinear dynamics-the case study of sea clutter. *Proc. IEEE*, **90**, 860–881.

Hendeby, G., Karlsson, R., and Gustafsson, F. (2007). A new formulation of the rao-blackwellized particle filter. In *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop on*, pages 84 –88.

Higuchi, T. (1997). Monte carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, **59**, 1–23.

Hol, J., Schön, T., and Gustafsson, F. (2006). On resampling algorithms for particle filters. In *Proc. IEEE Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK.

Hou, M. and Patton, R. (1998a). Input observability and input reconstruction. *Automatica*, **34**, 789–794.

Hou, M. and Patton, R. (1998b). Optimal filtering for systems with unknown inputs. *IEEE Trans. Automat. Contr.*, **43**, 445–449.

Jaeger, H. (2001). The echo state approach to analysing and training recurrent neural networks. GMD Report 148, Fraunhofer Institute for Autonomous Intelligent Systems (AIS), Schloss Birlinghoven 53757 Sankt Augustin, Germany.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. In D. Derickson, editor, *Advances in Neural Information Processing Systems 15*, pages 593–600. MIT Press, Cambridge, MA.

Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, **304**, 78–80.

Jin, L., Nikiforuk, P. N., and Gupta, M. M. (1995). Approximation of discrete time state space trajectories using dynamic recurrent neural networks. *IEEE Trans. Automat. Contr.*, **40**, 1266–1270.

Karlsson, R., Schön, T., and Gustafsson, F. (2005). Complexity analysis of the marginalized particle filter. *IEEE Trans. Signal Processing*, **53**, 4408–4411.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian non-linear state space models. *J. Comput. Graph. Statist.*, **5**(1), 1–25.

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, **93**, 1203–1215.

Krener, A. and Respondek, W. (1985). Nonlinear observers with linearizable error dynamics. *SIAM J. Control Optim*, **23**, 197–216.

Kushner, H. and Yang, J. (1995). Analysis of adaptive step-size sa algorithms for parameter tracking. *IEEE Trans. Automat. Contr.*, **40**, 1403 – 1410.

Liu, J. and West, M. (2001). *Combined parameter and state estimation in simulation-based filtering*. Sequential Monte Carlo in Practice. Springer-Verlag, New York.

Lockwood, M., Stamper, R., and Wild, M. (1999). A doubling of the sun's coronal magnetic field during the past 100 years. *Nature*, **399**, 437–439.

Madan, L. J., Gupta, M., and Nikiforuk, P. N. (1995). Universal approximation using dynamic recurrent neural networks: Discrete-time version. In *Proc. IEEE International Conference on Neural Networks (ICNN'95)*, pages 403–408, Montreal, Canada.

Minka, T. (2009). The lightspeed matlab toolbox. http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/.

Moon, T. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Mag.*, **13**, 47–60.

Natschlager, T., Maas, W., and Markram, H. (2002). The liquid computer: A novel strategy for real-time computing on time series. *Special Issue on Foundations of Information Processing of Telematik*, **8**, 39–43.

NGDC (2006). National geophysical data center; description of sunspot numbers. `http://www.ngdc.noaa.gov/stp/SOLAR/SSN/ssn.html`.

NGDC (2009). Annual international sunspot numbers available in the file. YEARLY.PLT, on the following link. `ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SUNSPOT_NUMBERS/`.

Nicolao, G., Sparacino, G., and Cobelli, C. (1997). Nonparametric input estimation in physiological systems: Problems, methods, and case studies. *Automatica*, **33**, 851–870.

Papoulis, A. (1985). Predictable processes and wold's decomposition: A review. *IEEE Trans. Acoust., Speech, Signal Processing*, **33**, 933–938.

Papoulis, A. and Pillai, S. (2002). *Probability, Random Variables and Stochastic Processes*, chapter 8. McGraw-Hill, fourth edition.

Pillonetto, G. and Bell, B. (2004). Deconvolution of non-stationary physical signals: a smooth variance model for insulin secretion rate. *Inverse Problems*, **20**, 367–383.

Pillonetto, G. and Bell, B. (2007). Bayes and empirical bayes semi-blind deconvolution using eigenfunctions of a prior covariance. *Automatica*, **43**, 1698–1712.

Pillonetto, G. and Saccomani, M. (2006). Input estimation in nonlinear dynamical systems using differential algebra techniques. *Automatica*, **42**, 2117–2129.

Roweis, S. and Ghahramani, Z. (2001). *Learning Nonlinear Dynamic Systems Using EM / Kalman Filtering and Neural Networks*, chapter 6. John Wiley & Sons, Toronto, ON.

Schön, T., Gustafsson, F., and Nordlund, P. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Trans. Signal Processing*, **53**, 2279–2289.

Schreiber, T. (1997). Detecting and analysing nonstationarity in a time series with nonlinear cross-predictions. *Phys. Rev. Lett.*, **78**, 843.

Schreiber, T. (1999). Interdisciplinary application of nonlinear time series methods. *Physics Reports*, **308**, 1–64.

Solanki, S., Schuessler, M., and Fligge, M. (2002). Secular variation of the sun's magnetic flux. *Astron. Astrophys.*, **383**, 706–712.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Trans. Signal Processing*, **50**, 281–289.

Szeliga, M., Verdes, P., Granitto, P., and Ceccatto, H. (2002). Extracting driving signals from non-stationary time series. In *Proc. Brazilian Symposium on Neural Networks (SBRN'97)*, pages 104–108, Pernambuco, Brazil.

Szeliga, M., Verdes, P., Granitto, P., and Ceccatto, H. (2003a). Artificial neural network learning of nonstationary behavior in time series. *Int J Neural Syst*, **13**, 103–109.

Szeliga, M., Verdes, P., Granitto, P., and Ceccatto, H. (2003b). Modeling nonstationary dynamics. *Physica A*, **197**, 190–194.

Tichavsky, P., Muravchik, C., and Nehorai, A. (1998). Posterior cramer-rao lower bounds for discrete-time nonlinear filtering. *IEEE Trans. Signal Processing*, **46**, 1386–1396.

Tikhonov, A. and Arsenin, V. (1977). *Solutions of ill-posed problems*, chapter 2. Winston/Wiley, Washington, DC.

Van Trees, H. (2001). *Detection, Estimation and Modulation Theory Part1*, chapter 2, pages 84–85. Wiley & Sons, New York, NY.

Verdes, P., Granitto, P., Navone, H., and Ceccatto, H. (2001). Nonstationary time-series analysis: Accurate reconstruction of driving forces. *Phys. Rev. Lett.*, **87**.

Verdes, P., Granitto, P., and Ceccatto, H. (2004). Secular behavior of solar magnetic activity: Nonstationary time-series analysis of the sunspot record. *Solar Physics*, **221**, 167–177.

Verdes, P., Granitto, P., and Ceccatto, H. (2006). Overembedding method for modeling nonstationary systems. *Physical Review Letters*, **118701**, 1–4.

Ward, K., Baker, C., and Watts, S. (1990). Maritime surveillance radar part 1: Radar scattering from the ocean surface. *IEE Proc.-F Rad., Sig. Proc.*, **137**, 51–62.

Xia, X. and Gao, W. (1989). Nonlinear observer design by observer error linearization. *SIAM J. Control Optim*, **27**, 199–216.

Zeidler, E. (1995). *Applied Functional Analysis: Applications to Mathematical Physics*. Springer, New York, NY.

Zhang, B., Chen, M., and Zhou, D. (2006). Chaotic secure communication based on particle filtering. *Chaos, Solitons & Fractals*, **30**, 1273–1280.

Zhang, B., Chen, M., Zhou, D., and Li, Z. (2007). Particle-filter-based estimation and prediction of chaotic states. *Chaos, Solitons & Fractals*, **32**, 1491–1498.

Zhu, F. and Han, Z. (2002). A note on observers for lipschitz nonlinear systems. *IEEE Trans. Automat. Contr.*, **47**, 1751–1754.

Zia, A., Reilly, J., Manton, J., and Shirani, S. (2007). An information geometric approach to ml estimation with incomplete data. *IEEE Trans. Signal Processing*, **55**, 3975–3986.

Zia, A., Kirubarajan, T., Reilly, J., Yee, D., Punithakumar, K., and Shirani, S. (2008). An em algorithm for nonlinear state estimation with model uncertainties. *IEEE Trans. Signal Processing*, **56**, 921–936.