KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

STRUCTURED KERNEL BASED MODELING AND ITS APPLICATION TO ELECTRIC LOAD FORECASTING

Promotoren :
Prof. dr. ir. B. De Moor
Prof. dr. ir. R. Belmans

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Marcelo ESPINOZA

June 2006

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

STRUCTURED KERNEL BASED MODELING AND ITS APPLICATION TO ELECTRIC LOAD FORECASTING

Jury:
Prof. dr. ir. E. Aernoudt, voorzitter
Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. R. Belmans, promotor
Prof. dr. ir. J. Suykens
Prof. dr. ir. J. Vandewalle
Prof. dr. W. Van Assche
Prof. dr. ir. J. Schoukens (VUB)
Prof. dr. J. Sjöberg (Chalmers)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Marcelo ESPINOZA

U.D.C. 681.5.015 June 2006

ISBN 90-5682-716-2

U.D.C. 681.5.015

D/2006/7515/50

To my Friends and Family

Foreword

To exist is to change, to change is to mature, to mature is to go on creating oneself endlessly.

- Henri Bergson (1859-1941).

The journey is the reward.

- Chinese Proverb.

This thesis is the product of my years as a research assistant in the SCD/SISTA research division of the Electrical Engineering Dept. of the Katholieke Universiteit Leuven. It has been a journey of learning and growth, packed with personal challenges and rewarding experiences. Having the opportunity of doing my doctoral research in this group, with many interesting colleagues and friends, in excellent working conditions shared in a relaxed atmosphere, has been a memorable experience for which I am very grateful.

I want to thank my promotor Bart De Moor for giving me the opportunity to join the group and start my doctoral research, and his continuous support since then. His interest and confidence in my work have been very important, as well as his support with respect to my funding and all the university paperwork required through these years. Through him, I also acknowledge all the help I received from the administrative and professional staff at SCD/SISTA. My gratitude goes as well to my co-promotor Ronnie Belmans, for his interest in my work since the first days of the ELIA project, and his active and detailed contributions to the manuscripts we prepared together. I also thank Johan Suykens for the daily discussions and his constructive support to my work. Many ideas developed in this thesis were generated as a result of our discussions. Many thanks to Joos Vandewalle, Walter Van Assche, Johan Schoukens and Jonas Sjöberg for accepting the invitation and

being part of the jury of this doctoral thesis. I am honored of having a jury of such excellent quality and renowned international reputation. Finally, to my friends and family, who helped me to hold the mind and heart up in moments of stress.

This work is the product of four years of research, cooperation, successful experiments and failed attempts. The obtained results are encouraging for further developments. It is with a sense of fulfillment that I invite you all to read this thesis.

Marcelo Espinoza

Leuven, June 2006.

Abstract

In the nonlinear system identification and forecasting of time-series, important challenges are in the accurate modeling by incorporation of prior knowledge and the estimation of such models from large scale datasets. In this thesis, the main scope is structured kernel based modeling and its application to electric load forecasting. We take as a starting point Least-Squares Support Vector Machines (LS-SVM) formulations for nonlinear regression. The primal-dual optimization framework can be extended to incorporate structured elements available from prior knowledge about the problem. The results are derived for the case of imposing symmetry to the estimated nonlinear model, imposing an additional parametric term for a new set of regressors and incorporating autocorrelation in the noise process of the regression. For each of these extensions, the goal is to include the additional structures in the form of equality constraints such that the resulting problem or subproblem remains convex, and Mercer's theorem can be applied with the use of a positive definite kernel and a kernel induced feature map. The prior information contained in the additional constraints becomes embedded at the kernel level, such that it can be used directly to evaluate the models at new datapoints. This property makes a contribution in terms of modularity of the model formulation, in the sense that different types of prior knowledge can be tested in practice simply by changing the kernel function being used. Furthermore, large scale versions of the different LS-SVM extensions can be formulated in primal space by using the Nyström method (which delivers finite dimensional approximations to the feature map as shown in the area of Gaussian processes) in the same way as for original fixed-size LS-SVMs. By considering each of the developed extensions as building blocks, a modular framework for the case of nonlinear system identification is further proposed. It is shown that this framework can be used for the estimation of NARX and AR-NARX model structures, with

different possible parameterization, exploiting the practical advantage of formulating the model in dual space and estimation in primal space for large sample sizes. The nonlinear system identification methods have been tested in a real-life industrial application by considering the short-term electricity load forecasting problem. Comparing different structures, we find that nonlinear models can capture the behavior of the load series and generate more accurate forecasts than the linear models, particularly when comparing not only black-box structures but also more structured representations. It is shown that the modular approach proposed in this thesis can be quite successful in the definition, estimation and final forecasting performance of nonlinear time series models.

Notation

Variables and Symbols

$\alpha, \beta, \gamma \in \mathbb{R}$	Scalar variables
$\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$	Vector variables
$\boldsymbol{Z}, \boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$	Matrix variables
$\boldsymbol{\Omega}_{ij}, \boldsymbol{\Omega} \in \mathbb{R}^{m \times n}$	Element at the i^{th} row and j^{th} column of A
$\boldsymbol{x}^T$	Transpose of the vector $\boldsymbol{x}$
$\boldsymbol{\Omega}^T$	Transpose of the matrix $\boldsymbol{\Omega}$
$[\boldsymbol{x}; \boldsymbol{z}], \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$	Stacked vectors : $[\boldsymbol{x}^T \ \boldsymbol{z}^T]^T \in \mathbb{R}^{n \times 2}$
$\mathscr{S}_M$	Sample of size M
$\{\boldsymbol{x}_i, y_i\}_{i=1}^N$	Sample of N datapoints
$\|x\|_2, x \in \mathbb{R}^n$	2-norm of a vector : $\sqrt{x^{\mathrm{T}} x}$
$K(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Kernel function evaluated on points $\boldsymbol{x}_i, \boldsymbol{x}_j$
$\boldsymbol{\varphi}(\cdot)$	Feature map
$\boldsymbol{I}$	Identity matrix
$\mathbf{1}$	Vector in which all components are equal to 1
$\hat{\boldsymbol{\varphi}}(\cdot)$	Finite dimensional approximation to the feature map
$\hat{\boldsymbol{\varphi}}_i(\cdot)$	$i-$th component of $\hat{\boldsymbol{\varphi}}$
$A(z^{-1})$	Polynomial on the lag operator
$\min_x$	Function minimization over x, optimal function value is returned
$\arg\min_x$	Function minimization over x, optimal value of x is returned
s.t.	Subject to constraints

Acronyms

LS-SVM	Least Squares Support Vector Machines
FS-LSSVM	Fixed-Size LS-SVM
SVM	Support Vector Machines
AR	Autoregression, Autoregressive (model)
ARX	Autoregressive model with eXogenous inputs
NARX	Nonlinear autoregressive model with eXogenous inputs
ARMA	Autoregression with moving average
ARIMA	Autoregression with integrated moving average
AR-ARX	ARX model with autoregressive residuals
AR-NARX	NARX model with autoregressive residuals
PL-NARX	NARX model parameterized with a partially linear structure
PL-AR-NARX	AR-NARX model parameterized with a partially linear structure
PAR	Periodic Autoregression
NFIR	Nonlinear Finite Impulse Response
SISO	Single Input Single Output
OLS	Ordinary Least Squares
RR	Ridge Regression
PCA	Principal Components Analysis
PLS	Partial Least Squares
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
STLF	Short Term Load Forecasting

Contents

Chapter 1

Introduction

To everything there is a season, and a time for every matter under the heaven; a time to be born, and a time to die; a time to plant, and a time to uproot what is planted.

- Ecclesiastes, 3:1-2.

The best way to predict the future is to create it.

- Peter F. Drucker (1905-2005).

If, for example, you come at four o'clock in the afternoon, then I shall begin to be happy at three o'clock.

- The Fox from *The Little Prince*.

1.1 General Background

The general scope of this thesis is related to the application of nonlinear regression techniques to time series data. Time series appear in virtually all human activities and natural phenomena. Many elements of the regressions models for time series may have a real-world interpretation, such as trends, growth rates, sensitivities, elasticities, and others. Moreover, the application of mathematical techniques for time series involves the notion of *prediction* or *forecasting*, being, according to the dictionary, "to know something before it happens". Since ancient times it has been recognized that knowing something before it happens usually has a benefit. Medieval merchants trading in crops and wool started to use rudimentary methods

for identification of cycles and patterns [73], as typically a prediction can be used to make appropriate manufacturing decisions, to trade based on it, to borrow, to lend, to invest, to buy, to sell, and more. Time series are usually represented in a graph with the time on the horizontal axis, the magnitude of the series on the vertical axis. Typically a model is estimated from available historical information up to a given moment in time. The model is then used to compute forecasts for the future, as shown in Figure 1.1.

With the development of statistical tools and estimation methods, forecasting techniques have been investigated for the last two hundred years. Pearson is often seen as one of the first to publish a rigorous treatment on correlation and regression in 1897. His work was based on studies of social impact, as that of many of his contemporaries. In the decades of 1920 and 1930, the seminal work of Yule [142] and Wold laid the ground for a powerful formalization of time series analysis, incorporating the random disturbance elements. Since then, the problem of forecasting has been tackled by researchers working in statistics, econometrics and system identification. Although there are differences between the working methodologies of each of these disciplines, they share the common philosophy of building sound mathematical methods in order to be able to predict the evolution of a variable, or a set of variables, for a given time in the future. Important developments in the area of time series prediction were obtained in the period 1940-1990 from all these disciplines: nonparametric estimation [53], maximum likelihood estimation [140], nonlinear mixing processes [46], unit-root tests and cointegration [26], seasonal analysis [64], the Box-Jenkins methodology [13], ARIMA models [51], Kalman filtering [70], model selection criteria [4], identification for control [74], and many others.

From a different starting point, the field of artificial intelligence provided new insights into forecasting methodologies. Work done [100] in the 1980s renewed interest in the problem of neural networks applied to pattern analysis, by presenting a learning algorithm for multilayer perceptrons, in a direct generalization of the perceptrons developed in the 1950s. Since then, neural networks started to be used as nonlinear forecasting tools in different applications. Later, in the 1990s, another major development took place. The framework developed by Vapnik [129] (Statistical Learning Theory) proposed a new view to the problem of data modeling, based on the concept of empirical risk minimization, leading to the development of

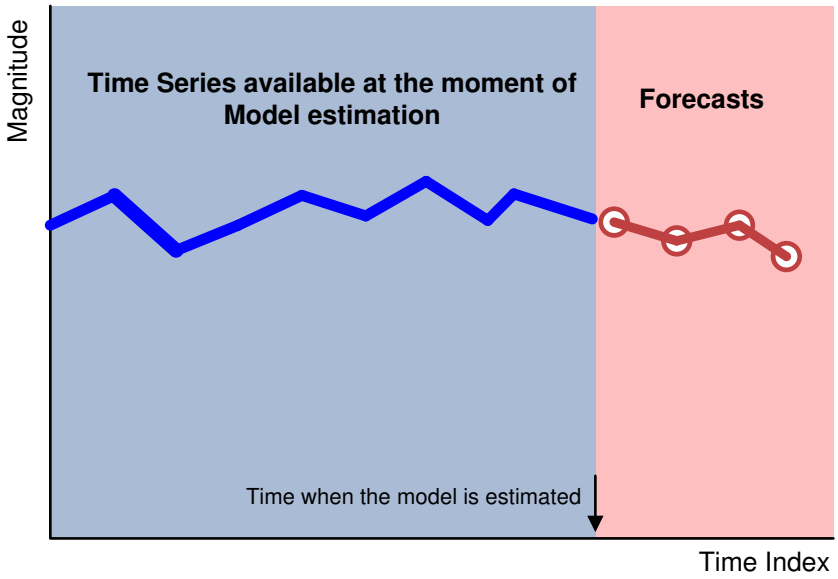

Figure 1.1: *Time series graphical representation. The available historical information is used to build a model, which is later used to produce forecasts for the future.*

Support Vector Machines (SVMs) and further kernel methods. Shortly after, the Least Squares Support Vector Machines (LS-SVM) technique was formulated [116] as a modified version of standard SVMs, aiming to a wider range of problems. In recent years, the field of system identification has adopted these nonlinear techniques [105], among others, as estimation methods for increasingly complex problems, focusing on issues like model structure definitions, parameterizations, implementability, and others.

1.2 Challenges and Problems

This work is related to nonlinear estimation techniques and their use in system identification for real-life problems. In this context, the current challenges in these areas can be summarized as follows.

Large Scale Problems. Technological advances in hardware for data

storage and computing power have led to the availability of large databases in industry and business, as found in banking, finance, process industry, and others. Large datasets pose a challenge to the techniques developed for modeling and forecasting. Building and estimating a model to be used with large datasets require, in many cases, specific implementations for cases where, otherwise, it would be almost unfeasible to estimate a model. Adaptive techniques, update mechanisms, subsampling, bagging, boosting, ensemble methods, are modeling strategies originating from the practical constraints raised by large scale problems. In this context, this thesis is oriented towards the implementation of nonlinear regression techniques that can handle large time series for industrial and business problems.

Nonlinear System Identification. In recent years, important advances in nonlinear system identification have taken place. The discussion has steadily moved from using black-box formulations towards the use of gray-box models. Black-box refers to the case where the user has no information about the structural form of the process under study, therefore only using input-output measurements to build a nonlinear model. In such case, the model is parameterized using neural networks, (LS)-SVMs, wavelets, or any other technique, and it is estimated by solving an optimization problem that minimizes the prediction errors. Depending on the technique, solving the optimization problem may not be straightforward, due to the existence of local-minima if the function is non-convex. When the user has some knowledge on the problem at hand, this knowledge should be imposed to the model, following the rule: "do not estimate what you already know". A model that contains some elements of prior knowledge yet still having a black-box part are usually called "gray-box". The way in which prior knowledge is imposed to the model varies depending on the type of structures being used. New distinctions on different "shades of gray" have also been proposed nowadays. The current state of the art considers the following properties to be desirable in the context of nonlinear system identification:

- Interpretability of the model. When the goal is not only to compute predictions, but also to understand the underlying relation between the input-output variables, it is desirable that the model can have a degree of interpretability. The parameters of a linear model can be directly interpretable in terms of sensitivities or elasticities. For the case of black-box or gray-box models, this is more difficult to achieve, as it depends on the particular estimation technique being used.

- Imposing prior knowledge to reduce model complexity. It is known that nonlinear black-box models are very general and powerful techniques. However, they might be too general when the user has prior knowledge on some parts of the process under study. It is important to have a modeling strategy that can incorporate these elements in a simple manner.

- Keeping the convexity of the problem. The advantage of working with a convex optimization problem is that there is a unique solution. Some black-box estimation methods do not have this property, making them prone to the problem of local-minima.

In this context, the main contributions of this thesis are built towards the above mentioned challenges.

Real-life applications. The time series considered in this thesis come from a real-life industrial problem. The use of forecasting methods has been particularly important in the energy sector. Electricity cannot be efficiently stored in large quantities, meaning that the quantity generated at any given time always has to cover all the demand by the final consumers, including grid losses. Forecasts of power load demand are used to decide if extra generation has to be provided by increasing the output of on-line generators or by committing one or more extra units. Similarly, forecasts are also used to decide if an already running generation unit should be decreased in output or even switched off. Moreover, the flow along the transmission lines is affected by the different generation profiles, possibly leading to congestion problems. On the other hand, the liberalization of the electric energy markets has led to the development of energy exchanges, where consumers, generators and traders can interact leading to price settings. In this respect also forecasts are extremely important. Large datasets from the Belgian Transmission System Operator ELIA are used in this thesis as examples to illustrate the importance and possibilities of the implementation of nonlinear forecasting techniques for the problem of short-term load forecasting. Building a model for load forecasting is not straightforward, due to the presence of seasonal patterns in different levels. There is a winter-summer pattern, a weekly pattern and an intra-daily pattern. Figure 1.2 shows an example of a load series, where the seasonal patterns are clearly visible. These different patterns also interact with other external variables that affect the load, the weather fluctuations being one of the most importants. When the weather is cold, there is a requirement

for heating which translates in an increase of the energy demand. Hot days in summer trigger the use of air conditioning equipment, also increasing the demand. The effect of weather in the load is nonlinear, which is one of the main reasons to use nonlinear models for this problem. However, for the purposes of long-term and mid-term planning, year-to-year comparisons and scenario analysis, it is important to have interpretable models. Particularly, the identification of the *normalized* yearly peak load is important for a correct estimation of the growth trends of the energy consumption [28]. A model has to be able to tell how much of the peak was due to the weather conditions of that particular day, so it can be corrected towards a normal meteorological year. Other types of analysis can be done by , for example, comparing the consumption of different regions or identifying customer profiles.

Most of the properties of the load series are also present in observations generated from other industrial or business activities. Utilities (gas, water) also share seasonal patterns and large records available. Traffic in highways is monitored and measured, with important peak hours and seasonal effects. Internet traffic, mobile communications, credit-card transactions, and others, also share some of the properties described above. The methods developed in this work may be useful in those contexts as well.

1.3 Objectives

This work takes elements from statistical learning theory and optimization, following the LS-SVM approach. It also covers elements from nonlinear system identification, by formalizing model structures to be identified with nonlinear regression methods; and finally it covers a real-life application problem. The structure of this thesis starts from theoretical contents, gradually working towards the application. In this context, the objectives of this work can be summarized as follows.

1. In the LS-SVM context, the objective is to extend the nonlinear regression formulation towards the inclusion of structured elements. Following the rule that you "do not estimate what you already know", the objective here is to include elements of prior knowledge from the problem at hand, in the form of additional constraints to the least-squares optimization problem being the central formulation

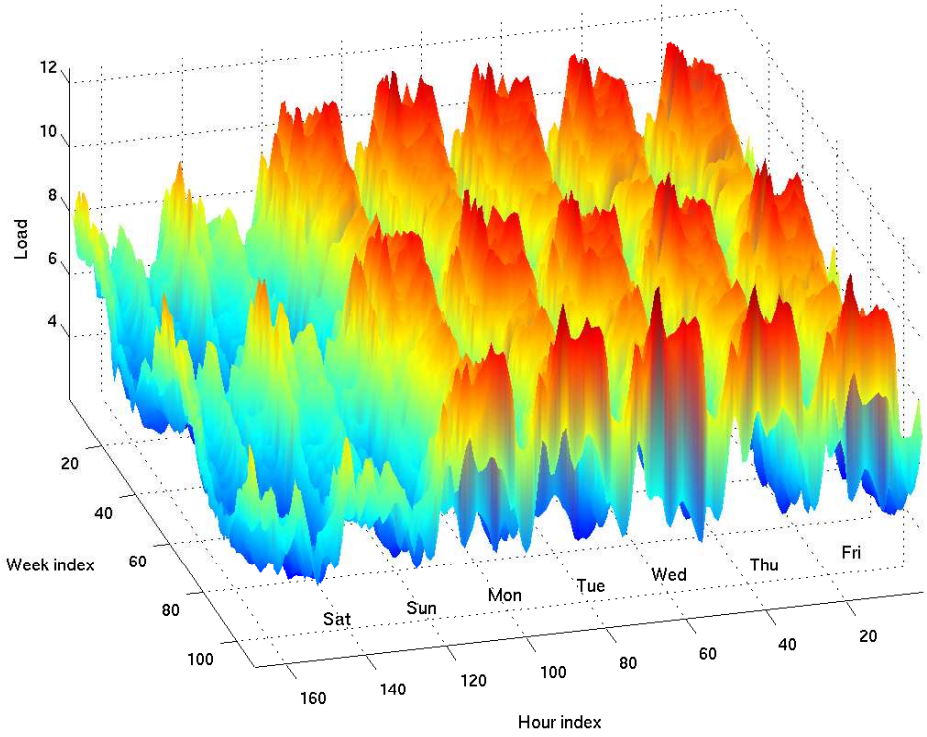

Figure 1.2: *Example of a load series where the seasonal patterns are clearly visible. The weekend is different from the working days. Every day has a peak in the morning, and another in the evening. The yearly cycle is visible when comparing the different week profiles.*

of LS-SVM. The goal is to extend the formulation by means of a modular approach, while maintaining the convexity of the optimization problem.

2. The second objective has to do with formalizing the link between nonlinear regression techniques and model structures in the context of nonlinear system identification. The goal is to define model structures, parameterizations and the corresponding estimation methods, exploiting the modular structure of the nonlinear regression techniques developed around LS-SVM.

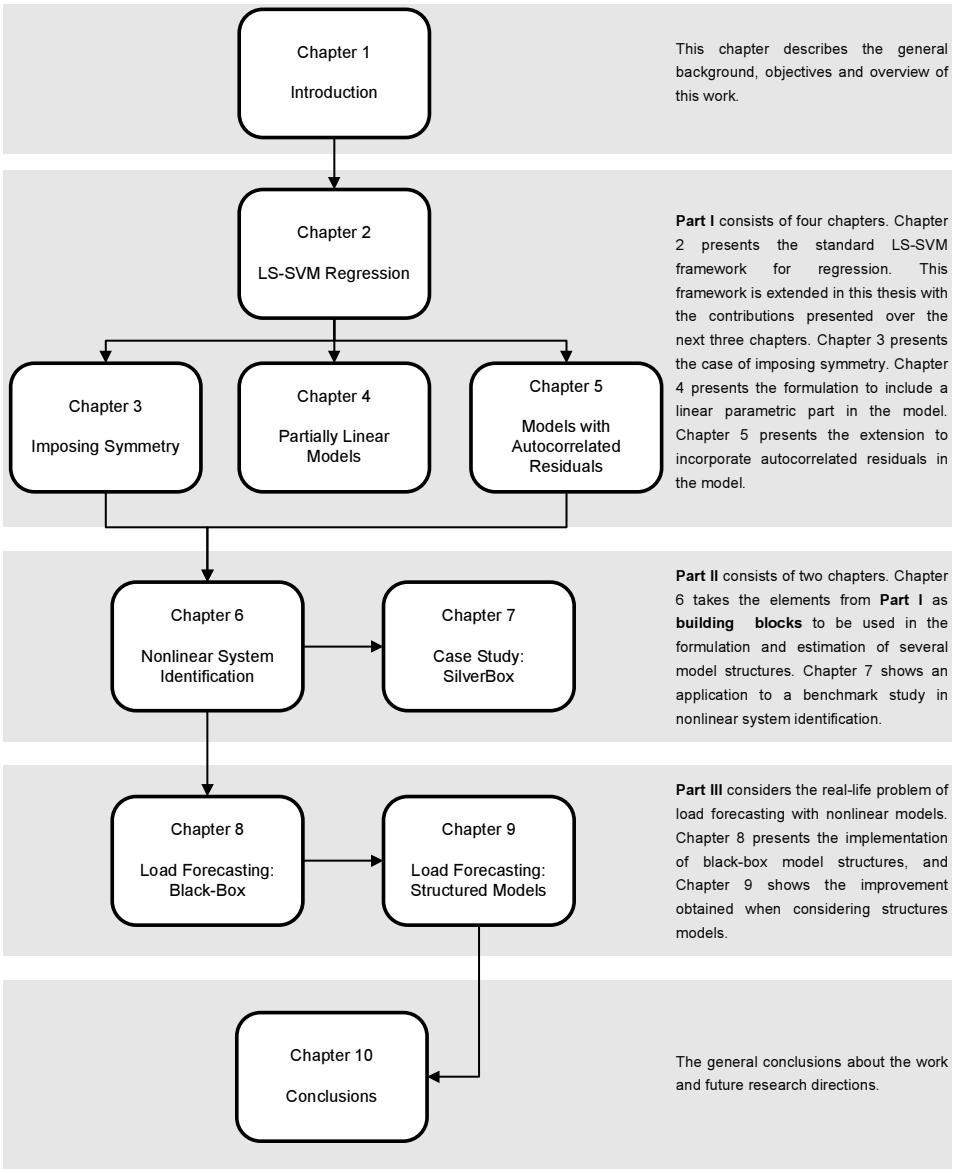

Figure 1.3: *The structure of the thesis. The arrows suggest the reading order of the chapters.*

3. The third objective is related to the real-life application of short-term load forecasting. Given different model structures and estimation methods, the goal is to build an implementation that can successfully produce accurate forecasts from the large available load series.

1.4 Chapter by Chapter Overview

The thesis is organized in 3 parts as shown on Figure 1.3. With the exception of Chapter 2, providing the setting for the thesis and providing the reader with the introduction to the LS-SVM, all subsequent chapters contain the different contributions of this work.

- **Estimation Techniques.** Part I is related to nonlinear regression techniques built from the LS-SVM formulation. It is a more theoretical part.
  - Chapter 2 presents the standard LS-SVM formulation for nonlinear regression [114]. The chapter starts with a basic description of kernel functions and Mercer's theorem, being one of the essential elements for a correct understanding of LS-SVMs. The chapter describes the LS-SVM estimation method in dual form, and the methodology for estimation in primal space achieved by using Nyström methods. Estimation in primal space is the basis for the implementations of large scale problems [31].
  - Chapter 3 extends the LS-SVM formulation to the case where it is known beforehand that the nonlinear function being estimated is symmetric. It describes the effect of imposing this prior knowledge as an additional constraint, giving rise to the definition of an equivalent kernel. It also explores the case of a "soft constraint" in which the symmetry property may not be exact [33]. Practical examples are presented.
  - Chapter 4 continues the extensions of the LS-SVM regression by considering a partially linear structure. It describes the inclusion of a linear parametric term to the LS-SVM regression, discussing links with related statistical techniques, and providing conditions for a unique representation of the linear part [32, 34]. It is shown that the solution of the problem is unique. Practical examples are given.

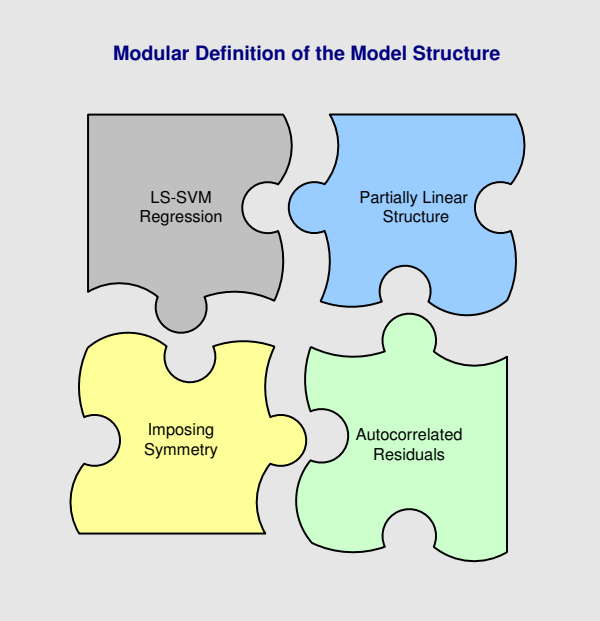

Figure 1.4: *Modular approach for nonlinear model structures definition. Elements from Part I are used as building blocks to define a model structure. The model is estimated in primal space for large scale problems, starting from a kernel matrix to be used in the Nyström approximation.*

  - Chapter 5 considers the case when the residuals of the LS-SVM regressions are autocorrelated. It starts with the formulation of a model where the residuals follow an AR(1) process [38]. It is shown that the correlation structure gets embedded into the kernel level, yielding a very straightforward estimation in primal space. The effect of the correlation of the residuals on the final predictor of the model is discussed. The chapter continues with the formalization of the general AR(q) case. In order to retain the convexity of the problem, the parameters of the AR(q) process are considered to be hyperparameters to be selected at another level (for example, using cross-validation, together with the kernel and regularization parameters). The chapter concludes with illustrative examples.

- **Nonlinear System Identification.** Part II takes the results of Part

I as building blocks, leading to a modular approach to model defintion and estimation, as shown on Figure 1.4. It provides a mixture of theory and practical implementations.

  - Chapter 6 moves towards the framework of nonlinear system identification. Using the modules from Part I, two model structures are defined: NARX and AR-NARX. Each of them is, in turn, parameterized either as a black-box nonlinear model, or as a partially linear model [39]. By combining the different alternatives, this chapter also provides the formalization of the estimation of a partially linear AR-NARX model (PL-AR-NARX), containing all structured elements from Part I in a single formulation. For each model structure, the chapter provides its equivalent kernel function, which is then used to formalize the estimation methods in primal space using the Nyström methods. Illustrative examples for large-scale chaotic time series [36] show the benefits of the techniques.
  - Chapter 7 presents the case-study of the so-called SilverBox dataset, based on a benchmark study of nonlinear regression methods [36]. The large dataset originates from a physical device, and the modeling strategy is defined in such a way that the generalization ability of the model is tested extensively. This chapter presents the implementation of nonlinear methods, including the symmetry and the partially linear extensions. It is shown that the performance of black-box models improves when structured elements are incorporated into the modeling stage.

- **Short Term Load Forecasting.** Part III implements the developed models in the context of load forecasting.

  - Chapter 8 provides the implementation of unstructured black-box models for load forecasting. It starts with a description of the available datasets and describes all practical steps towards building large scale black-box NARX models. It discusses the support vector selection by using quadratic Renyi entropy maximization, the approximation in primal space, the effect of having a sparse model, exploring different alternatives. In addition, extensive performance assessments are made with 10 different load series, comparing the performance of the nonlinear NARX model and the performance of a linear model estimated from the same set of information. The main contribution of this

chapter is the estimation of a nonlinear model using a sparse representation taking only 3% of the available sample to build the nonlinear mapping in primal space [37].

– Chapter 9 extends the analysis by considering the use of structured models, both linear and nonlinear. For the structured linear model, it describes the Periodic Autoregressive (PAR) method. For the structured nonlinear models, it considers the AR-NARX and PL-AR-NARX models as defined in Chapter 6. The estimation details, parameters interpretation and performance assessment are described. The performance assessment is based on 4 load series, and 50 different test datasets for each series. The main contribution is twofold. On the one hand, it is the first time that all of the structured models (linear and nonlinear) are applied to the load forecasting problem [29, 39]. On the other hand, it is shown that the use of partially linear model structures provides the best empirical tool for load forecasting, as they obtain a forecasting accuracy comparable to a fully nonlinear model, yet retaining a linear parametric part giving interpretability to the variables of interest. In addition, the properties of the PAR models are further exploited in Appendix A to provide a basis for clustering load profiles.

1.5 Contributions of this work

The main contributions of this work can be summarized as follows.

- **Imposing structured elements to LS-SVM regression.** Starting from the standard LS-SVM formulation for regression, structured elements have been added. We have considered cases where the structured elements can be described using additional equality constraints preserving the convexity of the problem. In this context, the studied structured elements are:

  – Imposing Symmetry to the LS-SVM regression. We have shown that it is possible to incorporate prior knowledge on the symmetry of the unknown nonlinear function to be identified with LS-SVMs. The symmetry, odd or even, is imposed as an additional constraint. Solving the problem in dual space yields an equivalent

kernel which embeds the information on symmetry. The case when symmetry is not exact is also studied. In this case, a second regularization term is included. It is shown that the LS-SVM regression can improve substantially its prediction performance when symmetry is imposed. The nonlinear model can identify correctly the unknown function even when some datapoints are missing, by using the symmetry information [Chapter 3, [33]].

- Formulation of a Partially Linear model with LS-SVM. The addition of a linear parametric part has been studied in the context of this work. It is shown that the model can have a unique solution under quite general conditions. A unique representation of the linear part is obtained when the linear and nonlinear parts have no common regressors. This model is particularly powerful, as it helps to reduce the complexity of the model, improving the generalization ability. Links with statistical techniques are studied [Chapter 4, [32, 34]].
- Including autocorrelated residuals in LS-SVM regression. Typically the residuals are assumed to be independent and identically distributed (i.i.d.). However, in the presence of correlated residuals, the LS-SVM regression is not able to identify correctly the unknown function because it captures the behavior of the function together with the structure of the residuals. We show that it is possible to build a nonlinear regression where the residuals follow an autoregressive process (AR). Moreover, the correlation structure gives a time dimension to a seemingly static problem. In order to preserve the convexity of the least-squares problem, the correlation parameters are considered to be hyperparameters, which are tuned in another level [Chapter 5, [38]].

- **Nonlinear System Identification using LS-SVMs.** In this context, the contributions of this work are mostly related to the definition and estimation of nonlinear model structures, and the implementation using fixed-size LS-SVM regression.

  - Definition of model structures using symmetry, partially linear models and autocorrelation in the residuals as building blocks. This modular approach has important practical advantages, as the user can plug-in particular elements containing prior-knowledge about the problem at hand. We have provided the

formulations in primal and dual space, with the corresponding equivalent kernel on each case [Chapter 6, [39]].

- Using LS-SVMs for chaotic time series prediction. It is known that chaotic time series are not predictable beyond a certain number of steps, which makes them a very hard test case for forecasting methodologies. In this context, we have applied LS-SVM regression, incorporating symmetry, to chaotic time series with excellent results [Chapter 6 [36]].
- Implementation of LS-SVM regression for the benchmark Silver-Box study. We have implemented several variants of the LS-SVM regression in primal space for a dataset containing more than 130,000 datapoints. This benchmark study was the basis of a Special Session in the NOLCOS conference in 2004. The simulation performance was the best among the results presented. In addition, the results are further improved when considering structured elements as partially linear models and/or symmetry [Chapter 7, [30], [34]].

- **Short-Term Load Forecasting.** In the real-life problem of load forecasting, the contribution of this work is related to the implementation and estimation of large-scale seasonal models for prediction.

  - Implementation of black-box (N)ARX models using fixed-size LS-SVM. We have shown that it is possible to build a nonlinear regression model with excellent prediction performance using a sparse representation based on less than 3% of the available dataset. The effect of the size of the sparse representation on the prediction performance is studied for 10 load series and compared to a linear ARX model [Chapter 9, [35, 37]].
  - Implementation of structured models. Using Periodic Autoregressive models (PAR), we have proposed a highly structured model of 24 equations for short-term forecasting. The model formulation is used as a template, being estimated individually for different series. The linear character of the model makes it possible to perform comparisons of the parameter estimates for the variables of interest [Chapter 10, [29], [28]]. An interesting by-product of the PAR models is that they allow to compute a Typical Daily Profile which is used in clustering of the load series [Appendix A, [29]]. Nonlinear structured models are also proposed. Particularly, the use of autocorrelated residuals

improve not only the prediction performance, but also the simulations performance on a 24 hours ahead basis. The use of partially linear models, in the context of load forecasting, provides an interesting way of obtaining interpretable results from the model [Chapter 10, [39]].

- **Other contributions.** Although not part of the main body of this thesis, other contributions from this work are:
  - Kernel based monotone regression. The case of a monotone nonlinear regression is studied, where the monotonicity is imposed using inequality constraints. This leads to a quadratic programming problem [93].
  - Clustering of load series using cepstral distances. Unlike the clustering done using the typical daily profile representation for each load series, here the clustering is done directly on the time series *models* (without building an explicit representation of each series) [12].
  - Energy islands modeling. Using genetic algorithms, the problem of defining size and placement of distributed generation units was addressed. Using different load profiles, a quasi-static optimization problem is formulated that takes into account transmission losses and seasonal changes [50].
  - Modeling of the glycemia-insulin dynamics in critically ill patients. We have explored ARX models to predict the glycemia of a patient at a given hour from the insulin, food and drugs administered by the nurses and medical doctors [127, 128].

Part I

Estimation Techniques

Chapter 2

Least Squares Support Vector Machines

The objective of this chapter is to present the LS-SVM formulation for regression [114] and to provide a description of the main conceptual aspects. Particularly, the primal-dual formulation and interpretation of LS-SVM makes it a powerful technique for different applications in nonlinear modeling. Belonging to the class of kernel methods, the LS-SVM is a modified version of the SVM where the optimization problem is reformulated using a least-squares specification with equality constraints. The original variables are mapped implicitly into a high (and possibly infinite) dimensional space based on Mercer's theorem. The LS-SVM has links with other kernel methods like Gaussian Processes [81,135], Reproducing Kernel Hilbert Spaces (RKHS) [133], regularization networks [40, 95], kernel ridge regression [101]. However, the LS-SVM framework based on convex optimization theory can be exploited further. In particular, the primal-dual structure of LS-SVM makes it possible to find a methodology for model estimation in primal space by using an explicit approximation of the nonlinear mapping. Based on the eigendecomposition of the kernel matrix and the use of Nyström techniques, it is possible to obtain a sparse approximation of the problem in primal space. The chapter is structured as follows. Section 2.1 introduces the notion of kernel functions and Mercer's theorem, which is

a cornerstone of the LS-SVM formulation, and discusses other kernel methods. Section 2.2 shows the derivation of the LS-SVM nonlinear regression where the solution is expressed in dual form. The estimation in primal space is presented in Section 2.3, where the approximation of the nonlinear mapping leads to the Fixed-Size LS-SVM.

2.1 Kernel Methods

Kernel methods are studied in different fields with several research directions. Although the methods are formulated in different ways, they all share the use kernel functions and the application of Mercer's theorem.

2.1.1 Kernel functions and Mercer's Theorem

A kernel may be characterized as a function from $X \times X$ to $\mathbb{R}$ (usually $X \subseteq \mathbb{R}^d$). A frequently used kernel function is the Radial Basis Function (RBF) kernel, shown in Figure 2.1, given by

$$K(\boldsymbol{x}, \boldsymbol{z}) = e^{-\frac{||\boldsymbol{x}-\boldsymbol{z}||_2^2}{\sigma^2}} \tag{2.1}$$

with σ a given coefficient.

Kernel functions have been used extensively in different domains, such as nonparametric estimation [53] and integral equation analysis. Since the 1990s, important developments from statistical learning theory [129] led to the incorporation of kernel functions into the class of so-called *kernel methods for pattern analysis*. Kernel methods include Support Vector Machines (SVMs, [129]) and Least-Squares Support Vector Machines (LS-SVM, [114]) among others.

The importance of kernel functions within the context of this work lies in allowing the computation of a model in a high (and possibly infinite) dimensional feature space, without having to compute explicitly the datapoints on the high dimensional space. This is illustrated in the following example for the case of a linear regression.

Consider the following simple linear regression to be estimated from a

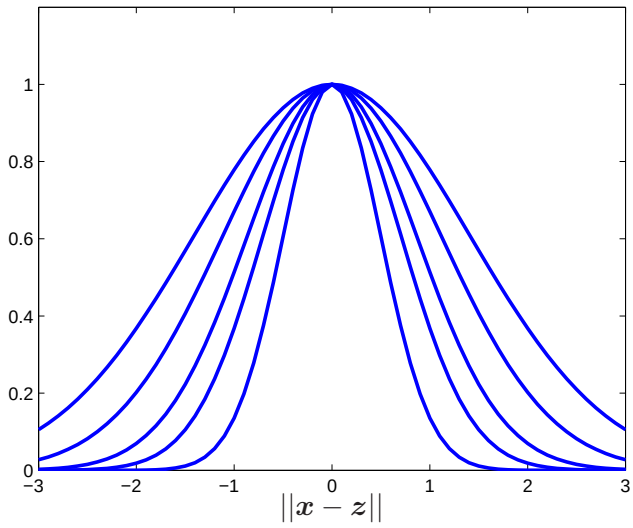

Figure 2.1: *RBF kernel function, shown here for different values of the parameter* σ.

training dataset $\{x_i, y_i\}_{i=1}^N$, with $y_i, x_i \in \mathbb{R}$,

$$y_i = wx_i + e_i, \tag{2.2}$$

where e_i is assumed to be independent and identically distributed (i.i.d.) with zero mean and constant variance. This linear regression can be estimated using least-squares, to obtain an estimate of $w \in \mathbb{R}$. If the system from which the dataset has been collected follows a linear process, the above regression (2.2) provides a good approximation of the system behavior. However, if the true system is given by

$$y = w_1 x^2 + w_2\sqrt{2}x + w_3, \tag{2.3}$$

the regression (2.2) is not correctly specified as it does not contain the nonlinear effect x^2. The correct regression to be estimated is, therefore,

$$y_i = w_1 x_i^2 + w_2\sqrt{2}x_i + w_3 + e_i. \tag{2.4}$$

In order to obtain the correct specification, the original input x has to be mapped to a higher dimensional space by means of the nonlinear mapping $\boldsymbol{\varphi}(x) : \mathbb{R} \to \mathbb{R}^3$

$$\boldsymbol{\varphi}(x) = [x^2, \sqrt{2}x, 1], \tag{2.5}$$

and then estimating the model

$$y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(x_i) + e_i. \tag{2.6}$$

In this example, the nonlinear mapping $\boldsymbol{\varphi}$ is assumed to be known, therefore the points in the high-dimensional space can be computed directly to arrive at the correct regression specification. However, it is possible to work with an unknown nonlinear mapping, using a relation between kernel functions and dot products in a Hilbert space. In the above example, it is possible to verify that the dot product of two vectors in the high-dimensional space is given by

$$\begin{aligned} \boldsymbol{\varphi}(x_1)^T \boldsymbol{\varphi}(x_2) &= [x_1^2, \sqrt{2}x_1, 1]^T [x_2^2, \sqrt{2}x_2, 1] && (2.7) \\ &= x_1^2 x_2^2 + 2x_1 x_2 + 1 && (2.8) \\ &= (x_1 x_2 + 1)^2, && (2.9) \end{aligned}$$

which is equivalent to the polynomial kernel

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \boldsymbol{x}_2 + c)^d, \tag{2.10}$$

evaluated at the points x_1, x_2 using $c = 1, d = 2$.

This simple example illustrates that it is possible to build a kernel function from a dot product of vectors in a high-dimensional space. However, the opposite is also possible. Starting from a kernel function, it is possible to obtain a high-dimensional space where the dot product is given by the kernel function evaluation. A kernel function can therefore *induce* a nonlinear mapping into a high dimensional space without explicitly computing it. This result was provided by James Mercer in 1909 working in the field of integral equations, in the form of the so-called Mercer's Theorem [88]:

Theorem 2.1. *(Mercer) Let X be a compact subset of $\mathbb{R}^n$. Suppose K is a continuous symmetric function such that the integral operator $T_K : L_2(X) \to L_2(X)$,*

$$(T_K f)(\cdot) = \int_X K(\cdot, \boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x},$$

is positive, i.e.,

$$\int_{X \times X} K(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{x}) f(\boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{z} \geq 0,$$

for all $f \in L_2(X)$. Then $K(\boldsymbol{x}, \boldsymbol{z})$ can be expanded in a uniformly converging series (on $X \times X$) in terms of T_K's eigenfunctions $\phi_j \in L_2(X)$, normalized

in such a way that $||\phi_j||_{L_2} = 1$ *and positive associated eigenvalues* $\lambda_j > 0$,

$$K(\boldsymbol{x}, \boldsymbol{z}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{z}).$$

■

The last summation can be written as

$$K(\boldsymbol{x}, \boldsymbol{z}) = \sum_{j=1}^{n_h} \sqrt{\lambda_j} \phi_j(\boldsymbol{x}) \sqrt{\lambda_j} \phi_j(\boldsymbol{z}), \tag{2.11}$$

where a mapping $\boldsymbol{\varphi} : \mathbb{R}^n \to \mathscr{H}$ can be defined with $\mathscr{H}$ a Hilbert space. Furthermore, it is possible to write $\boldsymbol{\varphi}_j(\boldsymbol{x}) = \sqrt{\lambda_j}\phi_j(\boldsymbol{x})$ and $\boldsymbol{\varphi}_j(\boldsymbol{z}) = \sqrt{\lambda_j}\phi_j(\boldsymbol{z})$ such that the kernel function can be expressed as the dot product

$$K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\varphi}(\boldsymbol{x})^T \boldsymbol{\varphi}(\boldsymbol{z}). \tag{2.12}$$

The application of (2.12) is called *kernel trick*, as first published in 1964 [2] in the context of pattern recognition. It means that any positive (semi) definite kernel function induces a nonlinear mapping to a higher (and possibly infinite) dimensional space, and provides the evaluation of the dot product in that space.

Remark 2.1. ["Feature" space]. *Usually the high dimensional space is called "feature space", and the nonlinear mapping* $\boldsymbol{\varphi}$ *"feature map" within the context of kernel methods, although sometimes they are referred to as the "hidden layer" using a multilayer perceptron interpretation. In the context of machine learning or datamining the terminology may differ. For example, in the case of clustering literature, the term "feature" refers to an input or variable, and the problem of variable selection is called "feature selection problem", which may lead to confusion when using kernel methods in those contexts. In this thesis, however, the term "feature space" is used to refer to the high dimensional space where the inputs are mapped to by means of the nonlinear mapping* $\boldsymbol{\varphi}$.

Remark 2.2. [Choices of Kernels] *For a positive definite kernel function* $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ *some common choices are:* $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$ *(linear kernel);* $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T \boldsymbol{x}_j + c)^d$ *(polynomial of degree* d*, with* $c > 0$ *a tuning parameter);* $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2/\sigma^2)$ *(RBF kernel), with* σ *a*

tuning parameter. On the other hand, the mapping $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x}$ *for* $\boldsymbol{x} \in \mathbb{R}^n$ *gives the linear kernel; the mapping* $\boldsymbol{\varphi}(x) = [1; \sqrt{2}x; x^2]$ *for* $x \in \mathbb{R}$ *gives the polynomial kernel of degree 2. The feature space related to the RBF kernel has been shown to be infinite dimensional [129]. The feature map may not be explicitly known in general. Taking a positive definite kernel guarantees the existence of the feature map. It is also possible to build kernels from kernels; e.g. a linear combination (using positive coefficients) of existing kernels is a valid kernel; a product of kernels is a valid kernel. For more information about building kernels from kernels, the reader is referred to [20].*

2.1.2 Different Kernel Methods

The work in this thesis is developed using the Least Squares Support Vector Machines (LS-SVMs [114]) nonlinear regression formulation as a basis technique. LS-SVMs and Support Vector Machines (SVMs) [95, 129, 136] follow the approach of a primal-dual optimization formulation, where both techniques make use of a so-called feature space where the inputs have been transformed by means of a (possibly infinite dimensional) nonlinear mapping $\boldsymbol{\varphi}$. This is converted to the dual space by means of Mercer's theorem and the use of a positive definite kernel, without computing explicitly the mapping $\boldsymbol{\varphi}$. The SVM model solves a quadratic programming problem in dual space, obtaining a sparse solution [20]. The LS-SVM formulation, on the other hand, solves a linear system under a least squares cost function with equality constraints, where the sparseness property can be obtained e.g. by sequentially pruning the support value spectrum.

Other directions in kernel methods follow different approaches. In Reproducing Kernel Hilbert Spaces (RKHS) [133] the problem of function estimation is treated as a variational problem ; Gaussian Processes (GP) [81,135] follow a probabilistic-Bayesian setting. Kriging [19] makes use of kernel methods in the context of spatio-temporal modeling with a strong probabilistic component. Although these different approaches have links with each other, e.g. for the simple case of static regression without a bias term it is well-known that GP, regularization networks and LS-SVM lead to the same set of linear equations to be solved at the dual level, in general the methodologies are different. Particularly, the primal-dual formulation of LS-SVM makes it easy to add additional constraints, which in the context of the present work makes it straightforward to incorporate more structure into the models, as it will be illustrated in subsequent chapters. In addition, the LS-SVM models

can be estimated in primal space directly with a sparse representation by using Nyström methods (which originated in the GP literature) for the case of large samples, which is also exploited in this thesis.

2.2 LS-SVM for Nonlinear Regression

The standard framework for LS-SVM estimation is based on a primal formulation which is solved in dual form. Given the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ the goal is to estimate a model of the form

$$y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i \tag{2.13}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$, $y \in \mathbb{R}$ and $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ is the mapping to a high dimensional (and possibly infinite dimensional) feature space, and the error terms e_i are assumed to be i.i.d. with zero mean and constant (and finite) variance.

The following optimization problem with a regularized cost function is formulated,

$$\min_{\boldsymbol{w},b,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^N e_i^2 \tag{2.14}$$

$$\text{such that} \quad y_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \quad i = 1, \ldots, N.$$

where γ is a regularization constant. The solution is formalized in the following lemma.

Lemma 2.1. *Given a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, the solution to (2.14) is given by the dual problem*

$$\left[\begin{array}{c|c} \boldsymbol{\Omega} + \frac{1}{\gamma}\mathbf{I} & \mathbf{1} \\ \hline \mathbf{1}^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \end{array}\right], \tag{2.15}$$

where $\boldsymbol{y} = [y_1, \ldots, y_N]^T, \boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, and $\boldsymbol{\Omega}$ is the kernel matrix with $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ $\forall i, j = 1 \ldots, N$.

Proof: Consider the Lagrangian of problem (2.14) $\mathscr{L}(\boldsymbol{w}, b, e_i; \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i - y_i)$, where $\alpha_i \in \mathbb{R}$ are

the Lagrange multipliers. The conditions for optimality are given by

$$\begin{cases} \frac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0 & \rightarrow \quad \boldsymbol{w} = \sum_{j=1}^{N} \alpha_j \boldsymbol{\varphi}(x_j) \\ \frac{\partial \mathscr{L}}{\partial b} = 0 & \rightarrow \quad \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial \mathscr{L}}{\partial e_i} = 0 & \rightarrow \quad \alpha_i = \gamma e_i, \qquad i = 1, \ldots, N \\ \frac{\partial \mathscr{L}}{\partial \alpha_i} = 0 & \rightarrow \quad y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \ i = 1, \ldots, N \end{cases} \tag{2.16}$$

With the application of Mercer's theorem [88] $\boldsymbol{\varphi}(\boldsymbol{x}_i)^T \boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with a positive definite kernel K, it is possible to eliminate $\boldsymbol{w}$ and e_i, obtaining $y_j = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b + \frac{\alpha_j}{\gamma}$. Building the kernel matrix $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and writing the equations in matrix notation gives the final system (2.15) ■

The final model is expressed in dual form

$$y(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b, \tag{2.17}$$

where it is not required to compute explicitly the nonlinear mapping $\boldsymbol{\varphi}(\cdot)$ as this is done implicitly through the use of positive definite kernel functions K. When the bias term is not present ($b = 0$), the result is equivalent to kernel ridge regression [101].

Remark 2.3. [Hyperparameter Selection] *Lemma 2.1 gives the solution of the LS-SVM regression estimation for a given kernel function K and a given regularization parameter γ. Usually training of the LS-SVM model involves an optimal selection of kernel parameters (σ for RBF kernel; c and d for a polynomial kernel) and the regularization term γ, which are typically denoted as* hyperparameters. *This is done in such a way that a good model performance is obtained. Figure 2.2 shows an example of a function approximation using LS-SVM with different kernel parameters. The original function (Top-left panel) is approximated with LS-SVM estimated on the available (noisy) points. By using different kernel parameters, clearly the quality of the approximation changes drastically. Therefore it is important to select optimal hyperparameters, using e.g. cross-validation techniques, or Bayesian inference [80, 94, 123–125].*

Remark 2.4. [LS-SVM and Mercer's Theorem] *The application of Mercer's theorem allows the final expression of the LS-SVM regression to be written in terms of the dual Lagrange multipliers. In the primal space, the LS-SVM regression can be viewed as a parametric model, while in the dual space*

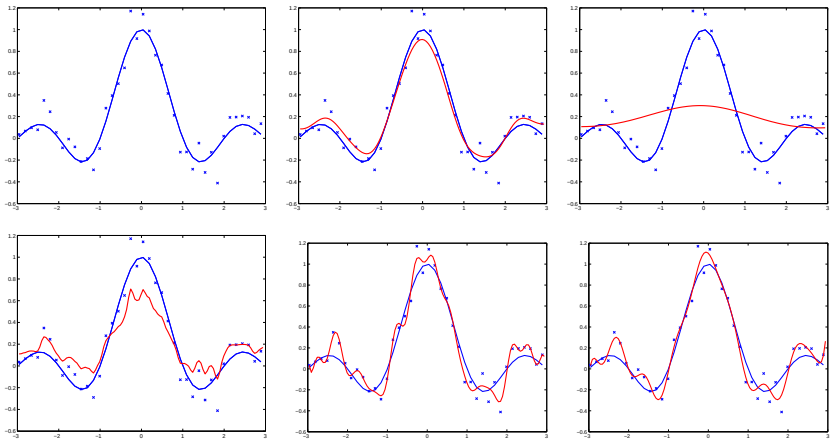

Figure 2.2: *Effect of using different kernel parameters for the same LS-SVM regression. The original function (Top-left) is approximated with an LS-SVM estimated on the available noise datapoints ('x'). Each one of the five different approximations is obtained with a different σ in the RBF kernel parameter. The selection of the optimal hyperparameters is important for a good performance of the model.*

it becomes non-parametric (the size of the solution vector grows with the number of data). This primal-dual formulation can be exploited further. The dimension of the system (2.15) is given by the number of datapoints, not by the dimension of the input vectors $\boldsymbol{x}$. This provides a practical advantage for working with small samples of high-dimensional inputs. On the other hand, when the available number of datapoints is too large, solving the system (2.15) can become too time consuming or simply unfeasible. Under these circumstances, it is possible to exploit the primal-dual formulation by finding an explicit approximation of the nonlinear mapping $\boldsymbol{\varphi}$ by means of Nyström methods. This concept is illustrated on Figure 2.2, and it is the subject of the next section.

Remark 2.5. [LS-SVM framework.] *In this work, the LS-SVMs are used for regression estimation. However, the LS-SVM framework is more general [114]. Given that the LS-SVM formulation works with equality constraints and an L_2 loss function, this optimization-based kernel methodology can be extended to a wide range of problems, including kernel versions of PCA (Principal Component Analysis) [115], FDA (Fisher Discriminant*

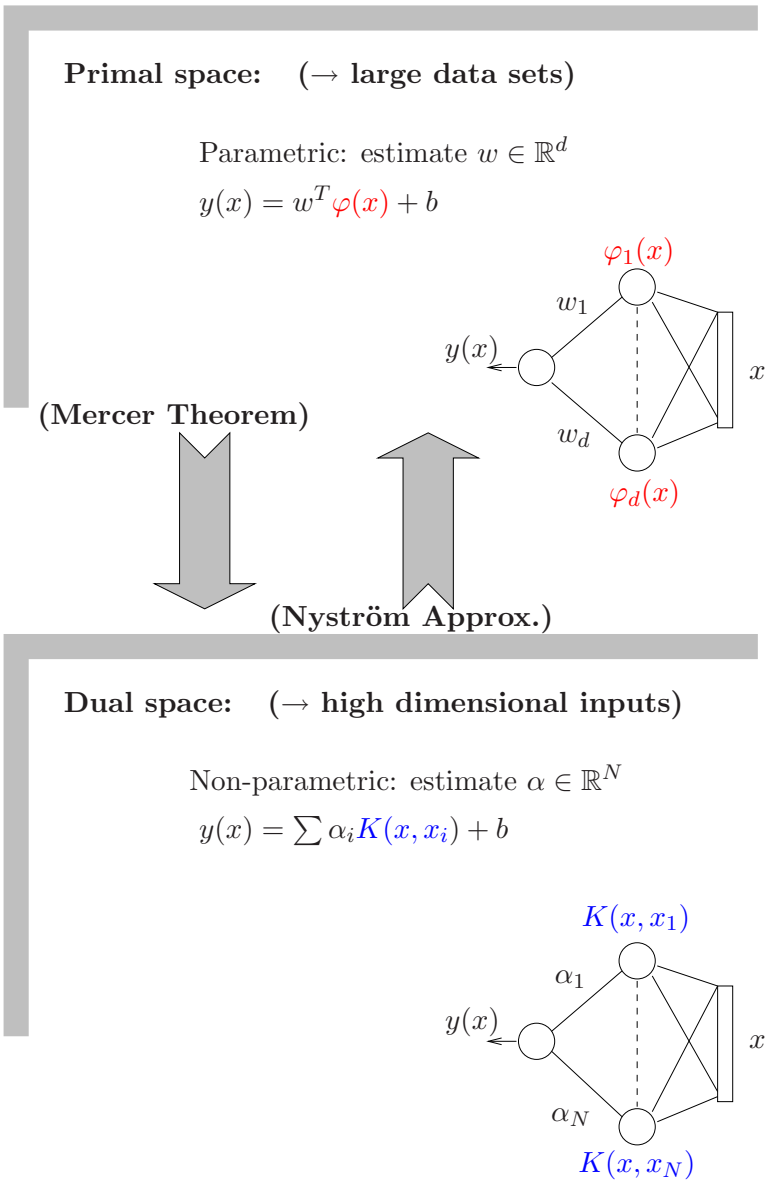

Figure 2.3: *LS-SVM primal-dual formulation for nonlinear regression. The final model can be written in dual form using Mercer's theorem ("kernel trick"). For the case of large scale problems, the model can be used in primal form by taking a finite dimensional approximation to the feature map φ by means of the Nyström technique. Figure taken from [114].*

Analysis) [122], CCA (Canonical Correlation Analysis) [126], PLS (Partial Least Squares) [60], spectral clustering [7], subspace methods, recurrent networks, control, and others. In general, LS-SVMs can be seen as a modular formulation from which more sophisticated kernel machines can be built using additional constraints or different loss functions for robustness or sparsity.

Remark 2.6. [Prediction Errors.] *With any modeling technique, it may be possible to obtain an estimation for the prediction error. In a linear regression, for example, it is possible to build confidence intervals around the predictions, starting from assumptions about the distributional properties of the residuals [51]. In the context of nonlinear modeling, however, extra assumptions may have to be taken. A Bayesian framework for LS-SVM regression in which the hyperparameters are given a prior distribution can produce prediction error bars, as developed in [114, 121]. Moreover, the quality of the prediction error estimation may be improved with dedicated methodologies. In the context of LS-SVM, bootstrapping methods have been studied for the LS-SVM regression [22]. However, this topic is outside the scope of this work, where the focus is on the incorporation of prior knowledge in the form of additional constraints without taking assumptions on distributional properties.*

2.3 Estimation in Primal Space

In this section, the estimation in primal space is described in terms of the explicit approximation of the nonlinear mapping $\boldsymbol{\varphi}$, and the implementation for a large scale problem.

2.3.1 Nyström Approximation in Primal Space

Explicit expressions for an approximation to $\boldsymbol{\varphi}$ can be obtained by means of an eigenvalue decomposition of the kernel matrix $\boldsymbol{\Omega}$ where $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}, \boldsymbol{x}_j)$. Given the integral equation

$$\int K(\boldsymbol{x}, \boldsymbol{x}_j)\phi_i(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \lambda_i \phi_i(\boldsymbol{x}_j), \tag{2.18}$$

with solutions λ_i and ϕ_i for a variable $\boldsymbol{x}$ with probability density $p(\boldsymbol{x})$, we can write

$$\boldsymbol{\varphi} = [\sqrt{\lambda_1}\phi_1, \sqrt{\lambda_2}\phi_2, \ldots, \sqrt{\lambda}_{n_h}\phi_{n_h}]. \tag{2.19}$$

Given the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, it is possible to approximate the integral by a sample average, as proposed in the context of Gaussian Processes [137,138]. This leads to the eigenvalue problem (Nyström approximation)

$$\frac{1}{N}\sum_{k=1}^{N} K(\boldsymbol{x}_k, \boldsymbol{x}_j)u_i(\boldsymbol{x}_k) = \lambda_i^{(s)} u_i(\boldsymbol{x}_j), \tag{2.20}$$

where the eigenvalues λ_i and eigenfunctions ϕ_i from the continuous problem can be approximated from the sample eigenvalues $\lambda_i^{(s)}$ and eigenvectors u_i as

$$\lambda_i = \frac{1}{N}\lambda_i^{(s)}, \phi_i = \sqrt{N}u_i. \tag{2.21}$$

Based on this approximation, it is possible to compute the eigendecomposition of the kernel matrix $\boldsymbol{\Omega}$ and use its eigenvalues and eigenvectors to compute the $i-$th required component of any point $\boldsymbol{x}$ (particularly those points not included in the original subsample) by means of

$$\hat{\boldsymbol{\varphi}}_i(\boldsymbol{x}) = \frac{N}{\sqrt{\lambda_i^{(s)}}}\sum_{k=1}^{N} u_{ki}K(\boldsymbol{x}_k, \boldsymbol{x}), \tag{2.22}$$

leading to the $M-$dimensional approximation

$$\hat{\boldsymbol{\varphi}}(\boldsymbol{x}) = [\hat{\boldsymbol{\varphi}}_1(\boldsymbol{x}), \hat{\boldsymbol{\varphi}}_2(\boldsymbol{x}), \ldots, \hat{\boldsymbol{\varphi}}_M(\boldsymbol{x})]^T. \tag{2.23}$$

This finite dimensional approximation $\hat{\boldsymbol{\varphi}}(\boldsymbol{x})$ can be used in the primal problem (2.14) to estimate $\boldsymbol{w}$ and b directly.

2.3.2 Sparse Approximations and Large Scale Problems

It is important to emphasize that the use of the entire training sample of size N to compute the approximation of $\boldsymbol{\varphi}$ produces a vector $\hat{\boldsymbol{\varphi}}(\boldsymbol{x})$ having N components, where each of them can be computed by (2.22) for all $\boldsymbol{x} \in \{\boldsymbol{x}_i\}_{i=1}^N$. However, for a large scale problem, it has been motivated [138] to use of a subsample of $M \ll N$ datapoints to compute $\hat{\boldsymbol{\varphi}}$. In this case, up

to M components are computed, leading to a sparse representation of the model when estimating in primal space [114]. The selection of the subsample of size M, the initial set of support vectors, is made *before* the estimation of the model, and the final performance of the model can depend on the quality of the initial selection. It is possible to take a random selection of M datapoints and use it to build the approximation of the nonlinear mapping $\boldsymbol{\varphi}$ as in [138], or to perform a more optimal selection based on quadratic Renyi entropy maximization as proposed in [114]. In this case, given a fixed-size M, the aim is to select the support vectors maximizing the quadratic Renyi entropy

$$H_R = -\log \int p(x)^2 dx \tag{2.24}$$

that, given the link between kernel PCA and density estimation established in [44], can be approximated by

$$\int \hat{p}(x)^2 dx = \frac{1}{N^2} \mathbf{1}^T \boldsymbol{\Omega} \mathbf{1}. \tag{2.25}$$

The use of this active selection procedure can be quite important for large scale problems, as it is related to the underlying density distribution of the sample. It is important to note that the performance of a model having an initial random selection of support vectors will be different from the performance of a model having an entropy-based selection depending on the characteristics of the dataset itself. A rather simple dataset may be approximated well by both methods; whereas in a more complex dataset, the models can show different performances. Intuitively, the initial selection should contain some important regions of the dataset, as it will be seen in the next chapters for the case of e.g. the Santa Fe Laser example [134].

The mechanism for selection of the initial support vectors by using the Renyi quadratic entropy works as follows [114].

1. Select a sample $\mathscr{S}_M$ of size M from the available data sample $\mathscr{S}_N$ of size N (typically $M \ll N$).
2. Compute a *small* kernel matrix $\boldsymbol{\Omega}_M$ using only the data from the subsample of size M:

$$\boldsymbol{\Omega}_{ij}^M = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{with} \quad \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathscr{S}_M$$

3. Evaluate the Renyi quadratic entropy using (2.24) and (2.25).

4. Select a datapoint $\boldsymbol{x}_1 \in \mathscr{S}_M$. Select another datapoint $\boldsymbol{x}_2$ from the remaining sample $\mathscr{S}_{N-M}$.

5. Exchange both points, and compute the entropy of the modified subsample $\mathscr{S}_M^* = \mathscr{S}_M \setminus \{\boldsymbol{x}_1\} \cup \{\boldsymbol{x}_2\}$.

6. If the entropy of $\mathscr{S}_M^*$ increases with respect to that of $\mathscr{S}_M$, the exchange is maintained. If the entropy does not increase, the exchange is not maintained and the datapoints are put back at their original positions.

7. Iterate from step 4 until convergence of the entropy magnitude is reached.

Remark 2.7. [Links with Kernel Principal Components Analysis] *It is interesting to note that (2.20) is related to applying kernel PCA [103, 115]. However, in our case the conceptual aim is to obtain an optimal finite dimensional approximation of the mapping $\boldsymbol{\varphi}$ in the feature space. Only in the case where the entire sample of size N is used for the approximation (i.e. $M = N$) then only (2.21) is computed and, therefore, the components of $\hat{\boldsymbol{\varphi}}$ are directly the eigenvectors of the kernel matrix $\boldsymbol{\Omega}$.*

2.3.3 Fixed-Size LS-SVM

Based on the explicit approximation $\hat{\boldsymbol{\varphi}}$ computed from an initial sample of M datapoints from the given dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, the Fixed-Size LS-SVM (FS-LSSVM) nonlinear regression estimator [114] can be formulated as follows:

$$\min_{\boldsymbol{w},b,\boldsymbol{e}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \tag{2.26}$$

$$\text{s.t.} \quad y_i = \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}(\boldsymbol{x}_i) + b + e_i, \quad i = 1, \ldots, N.$$

where γ is a regularization constant. Working with the explicit expression of $\hat{\boldsymbol{\varphi}}$ makes the problem (2.26) a ridge-regression problem, where the solution is given by the estimates of $\boldsymbol{w}$ and b. Solving the regression problem (2.26) can be done with traditional statistical techniques. Using $\gamma > 0$ is equivalent to ridge-regression [61]; using $\gamma = \infty$, to Ordinary Least Squares (OLS) [30,31]. For a discussion about the use of a regularization term and its properties in linear regression, the reader is referred to [11, 112, 113]. Given the fact

that $\hat{\boldsymbol{\varphi}}$ is finite dimensional, other estimators can be used for the parametric problem of estimating $\boldsymbol{w}, b$. This leads to fixed-size kernel methods.

The algorithm for the final implementation can be described by the following steps:

1. Consider the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$
2. Select a subsample of size M of the training points $\{\boldsymbol{x}_i\}_{i=1}^N$ using maximization of the quadratic Renyi entropy (2.25)
3. Use the selected subsample of size M to build a small kernel matrix $\boldsymbol{\Omega}_M$
4. Compute the eigenvectors u_i and eigenvalues $\lambda_i^{(s)}$ of $\boldsymbol{\Omega}_M$
5. Compute the approximation of the nonlinear mapping $\hat{\boldsymbol{\varphi}}(\boldsymbol{x}_i)$ using (2.22) for all points $i = 1, \ldots, N$
6. Solve the ridge regression problem (2.26) by eliminating e_i

Remark 2.8. [Equivalent Smoother matrix and Effective Number of Parameters]. *It is useful to write the vector of predictors from a model in the form*

$$\hat{\boldsymbol{y}} = S\boldsymbol{y},$$

where S denotes the smoother matrix. The effective number of parameters (degrees of freedom) is given by the trace of S [83]. In case the model is estimated with LS-SVM in dual space, and assuming the data has been centered, the smoother matrix takes the form

$$S = \boldsymbol{\Omega}(\boldsymbol{\Omega} + \gamma^{-1} I)^{-1},$$

with $\boldsymbol{\Omega}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. In case the model is estimated in primal space with Fixed-Size LS-SVM, the smoother matrix takes the form

$$S = (\hat{\boldsymbol{\Phi}}^T \boldsymbol{\Phi} + I\gamma^{-1})^{-1} \hat{\boldsymbol{\Phi}},$$

where $\hat{\boldsymbol{\Phi}}$ of dimension $N \times M$ is the matrix of regressors

$$\hat{\boldsymbol{\Phi}} = [\hat{\boldsymbol{\varphi}}(\boldsymbol{x}_1)^T; \hat{\boldsymbol{\varphi}}(\boldsymbol{x}_2)^T; \cdots, \hat{\boldsymbol{\varphi}}(\boldsymbol{x}_N)^T].$$

2.4 Example

Consider a static nonlinear modeling problem, with unidimensional input $\boldsymbol{x}$ and noisy target values $y_k = \text{sinc}(2\pi x_k) + e_k$, where

$$\text{sinc}(x) = \begin{cases} 1 & , \quad x = 0 \\ \frac{\sin(x)}{x} & , \quad \text{otherwise} \end{cases} \tag{2.27}$$

with e a white noise of variance 0.1 and $x \in [-0.5, 0.5]$. The sample size is $N = 200$, and the subsample for the fixed-size application will be selected with size $M = 20$.

- *Case I.* Using the full sample of size N to obtain the optimal hyperparameter, define the regressors and the final estimation.
- *Case II.* Using only a fixed-size subsample for finding the optimal hyperparameters, the regressors and the final model.

The results reported are:

1. The optimal σ found by minimizing the cross-validation MSE.
2. The MSE (mean squared error) both in-sample and out-of-sample.

The results are summarized in Table 2.1 and Figure 2.4. It is important to note that the good performance of the cases when $M \ll N$ is due not only to the quality of the Nyström approximation, but also to the good selection of the support vectors by means of the quadratic Renyi entropy maximization. For this example, the support vectors are quite uniformly distributed, as shown in Figure 2.4. The performance of the predictions for both Cases is also very good. For more complex datasets, it will be seen in Part II of this work that the support vectors are remarkably located in *important* regions of the dataset.

2.5 Conclusions

The Least Squares Support Vector Machines (LS-SVMs) formulation is a powerful nonlinear regression method. It builds a linear model in a high

	σ	M	MSE_{IN}	MSE_{OUT}
Case I	1.0	200	0.005	0.006
Case II	0.8	20	0.005	0.006

Table 2.1: *Performance of the estimations using LS-SVM in primal space for the Sinc function example. Case I uses of the full sample ($M = N$) to build the approximation in primal space, Case II makes use of a fixed-size LS-SVM ($M \ll N$) version.*

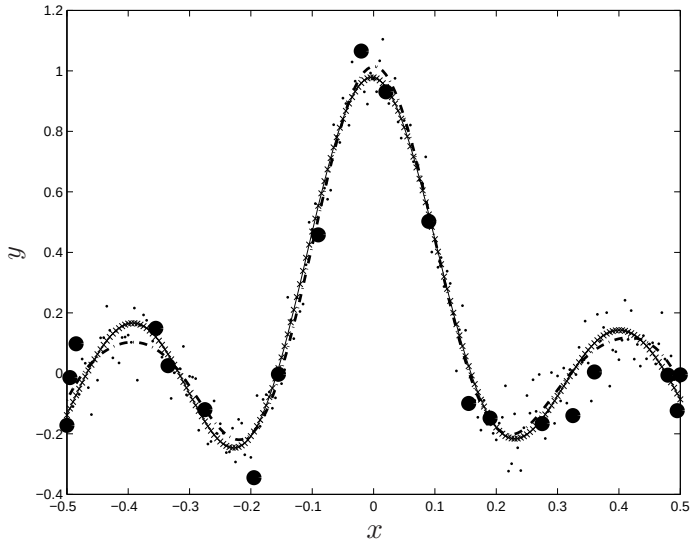

Figure 2.4: *Estimations for the noisy sinc function using LS-SVM in primal space. Case I ('-x' line) uses the full sample to build the approximation in primal space, Case II ('-.' line) uses only a small subsample of support vectors to build the approximation. The support vectors for Case II are depicted by the big dots.*

dimensional space where the inputs have been transformed by means of a (possibly infinite dimensional) nonlinear mapping $\boldsymbol{\varphi}$. This is converted to the dual space using Mercer's theorem and a positive definite kernel, without explicitly computing the mapping $\boldsymbol{\varphi}$. The LS-SVM formulation solves a linear system in dual space under a least-squares cost function, where sparseness can be obtained by e.g. sequentially pruning the support value spectrum or by means of a fixed-size subset selection approach. The LS-SVM training procedure involves the selection of a kernel parameter and the regularization parameter of the cost function, which can be done e.g. by cross-validation, Bayesian techniques or others.

The primal-dual formulation of the LS-SVM for regression can be exploited in order to obtain a sparse approximation using a finite dimensional approximation to the feature map $\boldsymbol{\varphi}$ with estimation in the primal space. The approximation is based on the Nyström method, which uses the eigendecomposition of the kernel matrix $\boldsymbol{\Omega}$ computed from a small sample of size $M \ll N$. This framework leads to the Fixed-Size LS-SVM, suitable for working with large datasets. The practical advantages of this estimation technique are shown in Part II, where the technique is used as an estimation method for a nonlinear system identification problem.

Chapter 3

Imposing Symmetry

One of the objectives of this work is to extend the LS-SVM formulation to incorporate prior knowledge in the estimation stage. This chapter presents one of the contributions of this thesis. It is shown how to use relevant prior information by imposing symmetry conditions (odd or even) to the Least Squares Support Vector Machines regression formulation. The symmetry property is found in many real-life applications, for example, in hysteresis curves in mechanical and ferromagnetic systems, in the behavior of chaotic systems, and others. The simple knowledge that a nonlinear function may show an even or odd symmetry can be imposed on the LS-SVM formulation in a straightforward way. This is done by adding a new constraint to the LS-SVM model, which finally translates into a new kernel. The equivalent kernel embodies the prior information on symmetry, and therefore the dimension of the final dual system is the same as in the unrestricted case. It is shown that using a regularization term and a soft constraint provides a general framework containing the unrestricted LS-SVM and the symmetry-constrained LS-SVM as extreme cases. Imposing symmetry can substantially improves the performance of the models, in terms of increasing the generalization ability and in reducing the model complexity. This chapter is structured as follows. Section 3.1 describes the derivation of the LS-SVM with symmetry constraints. Section 3.2 shows the case where the prior

information is imposed via a regularization parameter and a soft constraint. Illustrative examples are given in Section 3.3.

3.1 LS-SVM with Symmetry Constraints

The proposed inclusion of symmetry constraints (odd or even) to the nonlinearity within the LS-SVM regression framework can be formulated as follows. Given the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, with $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, the goal is to estimate a model of the form

$$y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \quad i = 1, \ldots, N, \tag{3.1}$$

where $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ is the mapping to a high dimensional feature space, and the error terms e_i are assumed to be i.i.d. with zero mean and constant (and finite) variance, and where the knowledge of symmetry (odd or even) is imposed on the nonlinear function as follows. A convex optimization problem with a regularized cost function is formulated:

$$\min_{\boldsymbol{w},b,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^N e_i^2 \tag{3.2}$$

$$\text{s.t.} \quad \begin{cases} y_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, & i = 1, \ldots, N, \\ \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) = a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_i), & i = 1, \ldots, N, \end{cases}$$

with $a \in \{-1, 1\}$ a given constant, taking either the value of 1 or -1 depending on the type of symmetry to be imposed. The first restriction is the standard model formulation in the LS-SVM framework. The second restriction is a shorthand notation for the cases where we want to impose the nonlinear function $\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i)$ to be even (resp. odd) by using $a = 1$ (resp. $a = -1$). The solution is formalized in the following lemma.

Lemma 3.1. *Given the problem (3.2) and a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ satisfying the assumption $K(\boldsymbol{x}_i, -\boldsymbol{x}_l) = K(-\boldsymbol{x}_i, \boldsymbol{x}_l)$ $\forall i, l = 1, \ldots, N$, the solution to (3.2) is given by the system*

$$\left[\begin{array}{c|c} \frac{1}{2}(\boldsymbol{\Omega} + a\boldsymbol{\Omega}^*) + \gamma^{-1}\mathbf{I} & \mathbf{1} \\ \hline \mathbf{1}^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \end{array}\right], \tag{3.3}$$

where $\boldsymbol{y} = [y_1, \ldots, y_N]^T$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\Omega}^*_{ij} = K(-\boldsymbol{x}_i, \boldsymbol{x}_j)$ $\forall i, j = 1, \ldots, N$.

Proof: Building the Lagrangian of the regularized cost function

$$\mathscr{L}(\boldsymbol{w}, b, e_i; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N}\alpha_i(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i - y_i) - \\ - \sum_{i=1}^{N}\beta_i(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) - a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_i)), \quad (3.4)$$

with $\alpha_i, \beta_i \in \mathbb{R}$ the Lagrange multipliers, and taking the optimality conditions, the following system of equations is obtained:

$$\begin{cases} \frac{\partial\mathscr{L}}{\partial\boldsymbol{w}} = 0 & \rightarrow & \boldsymbol{w} = \sum_{l=1}^{N}(\alpha_l + \beta_l)\boldsymbol{\varphi}(\boldsymbol{x}_l) - a\sum_{l=1}^{N}\beta_l\boldsymbol{\varphi}(-\boldsymbol{x}_l) \\ \frac{\partial\mathscr{L}}{\partial b} = 0 & \rightarrow & \sum_{i=1}^{N}\alpha_i = 0 \\ \frac{\partial\mathscr{L}}{\partial e_i} = 0 & \rightarrow & \alpha_i = \gamma e_i, \qquad i = 1, \ldots, N \\ \frac{\partial\mathscr{L}}{\partial\beta_i} = 0 & \rightarrow & \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) = a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_i), \qquad i = 1, \ldots, N \\ \frac{\partial\mathscr{L}}{\partial\alpha_i} = 0 & \rightarrow & y_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \qquad i = 1, \ldots, N \end{cases} \quad (3.5)$$

Using Mercer's theorem, $\boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ [114]. Under the assumption that $K(\boldsymbol{x}_i, -\boldsymbol{x}_j) = K(-\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall i, j = 1, \ldots, N$, the elimination of $\boldsymbol{w}, e_i$ and β_i yields

$$y_i = \frac{1}{2}\sum_{j=1}^{N}\alpha_j[K(\boldsymbol{x}_j, \boldsymbol{x}_i) + aK(-\boldsymbol{x}_j, \boldsymbol{x}_i)] + b + \frac{1}{\gamma}\alpha_i \quad (3.6)$$

and the final dual system can be written as

$$\left[\begin{array}{c|c} \frac{1}{2}(\boldsymbol{\Omega} + a\boldsymbol{\Omega}^*) + \frac{1}{\gamma}\mathbf{I} & \mathbf{1} \\ \hline \mathbf{1}^T & 0 \end{array}\right]\left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \end{array}\right], \quad (3.7)$$

with $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\Omega}^*_{ij} = K(-\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall i, j = 1, \ldots, N$. ■

Remark 3.1. [Equivalent Kernel] *The final model becomes*

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_i K_{eq}(\boldsymbol{x}_i, \boldsymbol{x}) + b. \quad (3.8)$$

where

$$K_{eq}(\boldsymbol{x}_i, \boldsymbol{x}) = \frac{1}{2}[(K(\boldsymbol{x}_i, \boldsymbol{x}) + aK(-\boldsymbol{x}_i, \boldsymbol{x})] \quad (3.9)$$

is the equivalent symmetric kernel that embodies the restriction on the nonlinearity. It is important to note that the final dual system (3.3) has the same size as the one obtained using the traditional unrestricted LS-SVM. Therefore, imposing the second constraint does not increase the size of the system to be solved, as the new information is translated to the kernel level. In addition, this regression can be estimated in primal space applying the Fixed-Size LS-SVM described in the previous chapter simply using the equivalent kernel function K_{eq} to build the approximation in (2.22).

Remark 3.2. [Validity of the Assumption] *The assumption $K(\boldsymbol{x}_i, -\boldsymbol{x}_j) = K(-\boldsymbol{x}_i, \boldsymbol{x}_j)$ $\forall i, j = 1, \ldots, N$ is easily verified for all kernel functions that can be expressed in terms of the distance between vectors, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$ (stationary kernels, for example, the RBF kernel) and those expressed in terms of the dot product $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K(\boldsymbol{x}_i^T \boldsymbol{x}_j)$ (nonstationary kernels, for example, the polynomial kernel), which are the most common kernel functions used in practice. From a theoretical point of view, in general the kernel function can be described by its spectral representation. For the general class of kernels for which the polynomial and RBF kernels are particular cases, the spectral representation can be written as [43]:*

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \int_{\mathbb{R}^p} \int_{\mathbb{R}^p} \cos(\boldsymbol{\theta}_1^T \boldsymbol{x}_i - \boldsymbol{\theta}_2^T \boldsymbol{x}_j) F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \tag{3.10}$$

where F is a bounded symmetric measure. In this representation, noticing that $\cos(z) = \cos(-z)$, it is easy to verify the required assumption:

$$\begin{aligned} K(\boldsymbol{x}_i, -\boldsymbol{x}_j) &= \int_{\mathbb{R}^p} \int_{\mathbb{R}^p} \cos(\boldsymbol{\theta}_1^T \boldsymbol{x}_i + \boldsymbol{\theta}_2^T \boldsymbol{x}_j) F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \\ &= \int_{\mathbb{R}^p} \int_{\mathbb{R}^p} \cos(-[-\boldsymbol{\theta}_1^T \boldsymbol{x}_i - \boldsymbol{\theta}_2^T \boldsymbol{x}_j]) F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \\ &= \int_{\mathbb{R}^p} \int_{\mathbb{R}^p} \cos(-\boldsymbol{\theta}_1^T \boldsymbol{x}_i - \boldsymbol{\theta}_2^T \boldsymbol{x}_j) F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \\ &= K(-\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned}$$

Therefore, for a large class of kernels, most used in practice for nonlinear system identification, the required assumption holds. However, this may not be a general property for all possible kernels, especially those still to be defined in new applications fields (for example, text patterns, chemical molecules, graphs, etc.).

3.2 Imposing Symmetry via a Regularization Term

In this section we propose to impose symmetry as a soft constraint, which can be interpreted as a weak prior knowledge. Under the same definitions for the initial dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ and the model formulation, now the following optimization problem with a regularized cost function is formulated:

$$\min_{\boldsymbol{w},b,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma_1\frac{1}{2}\sum_{i=1}^N e_i^2 + \gamma_2\frac{1}{2}\sum_{i=1}^N r_i^2 \tag{3.11}$$

$$\text{s.t.} \begin{cases} y_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, & i = 1,\ldots,N, \\ \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) = a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_i) + r_i, & i = 1,\ldots,N, \end{cases}$$

with $a \in \{-1, 1\}$ a given constant. The second restriction, imposing $\boldsymbol{w}^T\boldsymbol{\varphi}(x_i)$ to be even (resp. odd) by using $a = 1$ (resp. $a = -1$), contains a residual term r_i allowing the restriction not to be exact. The "fitting" of this second restriction is included in the cost function via a new regularization constant γ_2. The solution is formalized in the following lemma.

Lemma 3.2. *Given a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ satisfying $K(\boldsymbol{x}_i, -\boldsymbol{x}_j) = K(-\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall i, j = 1,\ldots,N$, the solution of the problem (3.11) is given by the system*

$$\left[\begin{array}{c|c} \boldsymbol{\Omega}_{eq} + \gamma_1^{-1}\mathbf{I} & \mathbf{1} \\ \hline \mathbf{1}^T & 0 \end{array}\right]\left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \end{array}\right], \tag{3.12}$$

where

$$\boldsymbol{\Omega}_{eq} = \frac{1}{2}(\boldsymbol{\Omega} + a\boldsymbol{\Omega}^*) + \frac{1}{2\gamma_2}(a\boldsymbol{\Omega}^* - \boldsymbol{\Omega} + \frac{1}{2\gamma_2}\boldsymbol{I})^{-1} \tag{3.13}$$

and $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\Omega}^*_{ij} = K(-\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall i, j = 1,\ldots,N$.

Proof: Building the Lagrangian as in (3.4) and taking the optimality conditions, we obtain the system

$$\begin{cases} \frac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0 & \to \quad \boldsymbol{w} = \sum_{l=1}^N(\alpha_l + \beta_l)\boldsymbol{\varphi}(\boldsymbol{x}_l) - a\sum_{l=1}^N \beta_l\boldsymbol{\varphi}(-\boldsymbol{x}_l) \\ \frac{\partial \mathscr{L}}{\partial b} = 0 & \to \quad \sum_{i=1}^N \alpha_i = 0 \\ \frac{\partial \mathscr{L}}{\partial e_i} = 0 & \to \quad \alpha_i = \gamma_1 e_i, \qquad i = 1,\ldots,N \\ \frac{\partial \mathscr{L}}{\partial r_i} = 0 & \to \quad -\beta_i = \gamma_2 r_i, \qquad i = 1,\ldots,N \\ \frac{\partial \mathscr{L}}{\partial \beta_i} = 0 & \to \quad \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) = a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_i) + r_i, \qquad i = 1,\ldots,N \\ \frac{\partial \mathscr{L}}{\partial \alpha_i} = 0 & \to \quad y_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \qquad i = 1,\ldots,N. \end{cases} \tag{3.14}$$

From this system, one can express a relation between the vectors of Lagrange multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as

$$(\boldsymbol{\Omega} - a\boldsymbol{\Omega}^*)\boldsymbol{\alpha} = (2a\boldsymbol{\Omega}^* - 2\boldsymbol{\Omega} + \frac{1}{\gamma_2}\boldsymbol{I})\boldsymbol{\beta} \tag{3.15}$$

On the other hand, the elimination of $\boldsymbol{w}$ and e_i using the optimality conditions gives, in matrix notation,

$$\boldsymbol{y} = \boldsymbol{\Omega}\boldsymbol{\alpha} + \boldsymbol{\Omega}\boldsymbol{\beta} - a\boldsymbol{\Omega}^*\boldsymbol{\beta} + \mathbf{1}b + \frac{1}{\gamma_1}\boldsymbol{\alpha} \tag{3.16}$$

Substituting $\boldsymbol{\beta}$ in terms of $\boldsymbol{\alpha}$ by (3.15) in (3.16) gives the final system (3.12). ■

Remark 3.3. [Role of second regularization term] *Imposing symmetry as a soft constraint gives rise to a new equivalent kernel*

$$\boldsymbol{\Omega}_{eq} = \frac{1}{2}(\boldsymbol{\Omega} + a\boldsymbol{\Omega}^*) + \frac{1}{2\gamma_2}(a\boldsymbol{\Omega}^* - \boldsymbol{\Omega} + \frac{1}{2\gamma_2}\boldsymbol{I})^{-1} \tag{3.17}$$

equal to the equivalent kernel of Section 3.1 when $\gamma_2 \to \infty$. This means that the hard constrained case of Section 3.1 is a particular case of the soft constrained derivation. In addition, if $\gamma_2 \to 0$ the regularized cost function from (3.11) becomes the cost function of the standard LS-SVM (2.14). When $\gamma_2 \to 0$ working with the soft constraint, the optimality condition related to r_i gives $\beta_i = 0$ thus eliminating the effect of the second constraint. Therefore, imposing symmetry via a regularization parameter and a soft constraint covers a continuum of cases: from the standard unconstrained LS-SVM ($\gamma_2 \to 0$, no prior knowledge) to the hard constrained case of Section 2 ($\gamma_2 \to \infty$, absolute prior knowledge). From this perspective, the regularization term γ_2 can measure the degree by which symmetry can be imposed. This is also related to the Bayesian framework where prior information can be imposed via a regularization term [82, 114].

3.3 Examples

In this section, some examples of the effects of imposing symmetry to the LS-SVM are presented. In all cases, an RBF kernel is used and the parameters σ and γ are found by 10-fold cross validation over the corresponding training sample. In each example, the results using the standard LS-SVM (that is,

full black-box model) are compared to those obtained using the symmetry-constrained LS-SVM (S-LS-SVM) (3.2). The examples are defined in such a way that there are not enough training datapoints in every region of the relevant space; thus, it is very difficult for a black-box model to"learn" the symmetry just by using the available information. The examples are compared in terms of complexity (effective number of parameters [132]), performance in the training sample (cross-validation mean squared error, MSE-CV) and generalization performance (MSE out of sample, MSE-OUT). The results are shown on Table 3.1.

Cubic function. The model to be identified is $y_k = x_k^3 + e_k$, where e_k is drawn from a normal distribution with zero mean and variance 0.2. The training data for this example consists of $x_k \in [0, 3]$ in increments of 0.1, containing only positive values. The goal is to observe how well the model generalizes to the negative values of x_k. The model is formulated simply as $y_k = \boldsymbol{\varphi}(x_k) + e_k$ to be identified by standard LS-SVM and by S-LS-SVM, where the symmetric condition is implemented using $a = -1$ in (3.2) (odd function). Figure 3.1 shows the performance of the estimated models. It is not surprising that the S-LS-SVM can generalize better by using the symmetry information from the problem at hand. The effective number of parameters is reduced from 4.4 (LS-SVM) to 3 (S-LS-SVM).

Sinc function in 2-D. The model to be identified is $y_k = 0.5[\mathrm{sinc}(x_k) + \mathrm{sinc}(z_k)] + e_k$, where e_k is drawn from a normal distribution with zero mean and variance 0.1. Training values for x_k range from -2.9 to 2.9, while the training values for z_k only take positive values in the range 0 to 2.9. The black box model is formulated as $y_k = \boldsymbol{\varphi}(x_k, z_k) + e_k$ and is estimated by LS-SVM and S-LS-SVM. The final models are then used to generalize to the other half of the space, where the input z_k is negative. The inclusion of a symmetry constraint ($a = 1$) exploits the prior knowledge that the problem is symmetric and it can extrapolate correctly, as shown in the bottom panel of Figure 3.2. In this case, the effective number of parameters is reduced from 29 to 25.

3.4 Conclusions

We have proposed and shown how to impose simple constraints from prior information on the symmetry of the unknown nonlinear function to be identified using LS-SVM. The constraint with the symmetry condition (odd

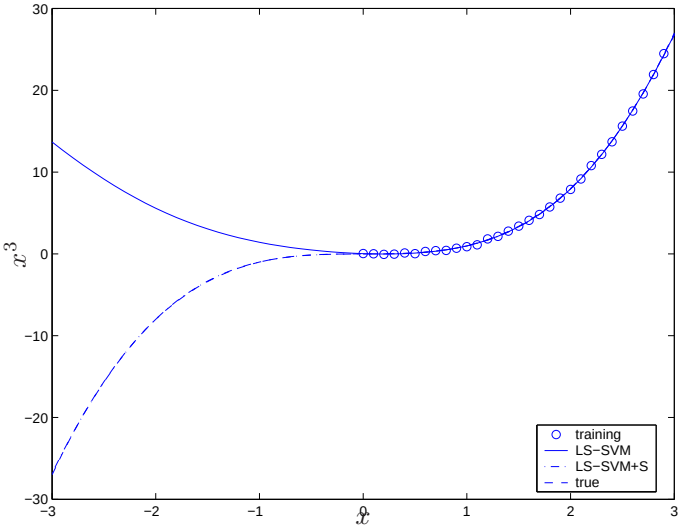

Figure 3.1: *Example of a cubic function approximation imposing the symmetry property to the LS-SVM. Training points and predictions with LS-SVM (thin line), Symmetric LS-SVM (dot-dashed) and the actual values (dashed line).*

	1-D Cubic	2-D Sinc
LS-SVM		
N_{eff}	4.4	29
MSE-CV	0.011	0.010
MSE-OUT	156.2	0.027
LS-SVM with Symmetry Constraint		
N_{eff}	3.0	25
MSE-CV	0.009	0.008
MSE-OUT	0.006	0.001

Table 3.1: *Performance comparison between LS-SVM and S-LS-SVM for two examples. The model that includes symmetry exploits the prior knowledge of the problem and it is able to extrapolate to the region where there are no training points.*

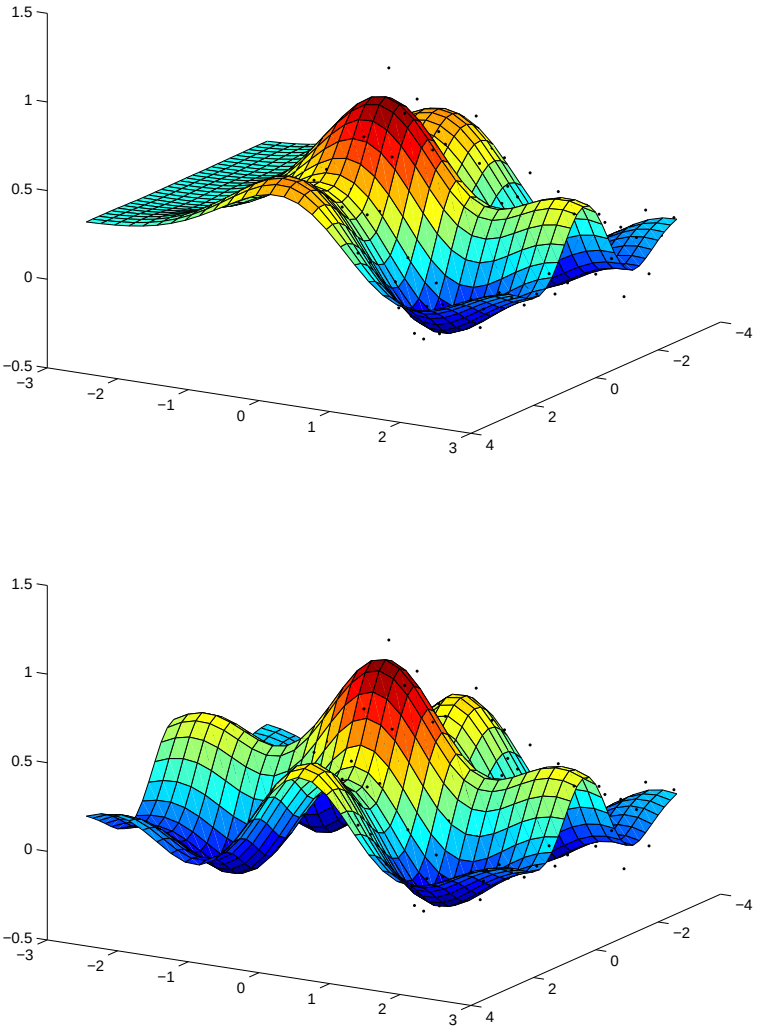

Figure 3.2: *Example of a sinc function surface approximation imposing symmetry to LS-SVM. Training points and predicted surface with LS-SVM (Top) and Symmetric LS-SVM (Bottom). The Symmetric LS-SVM can extrapolate correctly to the region where there are no training points given the prior knowledge that the function is symmetric.*

or even) translates into an equivalent kernel. This makes the dimension of the constrained dual system to remain equal to the unrestricted case. Imposing prior knowledge as a hard constraint is a straightforward extension of the LS-SVM, where the new kernel embodies the prior information. When symmetry is imposed as a soft constraint, the associated regularization term can be interpreted as the indicator up to which extent prior knowledge can be imposed. When this regularization term goes to infinity, the hard constraint case is recovered. When it goes to zero, the standard LS-SVM is recovered. Practical examples of imposing symmetry show satisfactory results. The benefits of imposing symmetry within the context of nonlinear system identification will be illustrated in Part II of this thesis.

Chapter 4

Partially Linear Models

In this chapter, another contribution of this work is presented. The LS-SVM formulation is extended to define a Partially Linear LS-SVM in order to estimate a regression containing a linear part and a nonlinear component. The fully nonlinear LS-SVM regression may be too general in some situations when there are reasons to include a linear part in the model. Furthermore, the goal of a specific problem may be to identify a linear part based on first-principles, while including a nonlinear black-box part in order to keep the overall model accuracy within satisfactory limits. In these cases it is desirable to have a technique that can lead to the estimation of a regression containing both a linear and a nonlinear structure [106]. Within the statistical literature, so-called "partially linear models" [54, 98, 107] have been developed since the mid-80s. These models contain a linear parametric part and also a nonparametric component estimated using (local) smoothing techniques, usually restricted to low dimensional input vectors [52]. The concept can be extended to the LS-SVM framework by defining a model for which the LS-SVM can capture a nonlinear component while a parametric linear part can be simultaneously identified, allowing the inclusion of large dimensional input vectors for the nonlinear part. For a given kernel, a unique solution exists when the parametric part has full column rank, although identifiability problems can arise for certain structures.

The solution has close links with traditional semiparametric techniques from the statistical literature. The properties of the model are illustrated by Monte Carlo simulations. This chapter is organized as follows. In Section 4.1 the Partially Linear LS-SVM is developed. Links with statistical techniques and properties of the solution are given in Section 4.2. Practical applications are reported in Section 4.3.

4.1 Partially Linear LS-SVM

Consider the following partially linear regression structure

$$y_i = \boldsymbol{\beta}^T \boldsymbol{z}_i + f(\boldsymbol{x}_i) + e_i, \qquad i = 1, \ldots, N, \tag{4.1}$$

where $\boldsymbol{z}_i \in \mathbb{R}^p, \boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{x}_i \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}$ is an unknown nonlinear function. The terms e_i are assumed to be i.i.d. random errors with zero mean and constant variance. To avoid identifiability problems, it is assumed that the variables $\boldsymbol{z}$ are not identical to $\boldsymbol{x}$, and in general, that $\boldsymbol{z}$ can not be mapped to $\boldsymbol{x}$ [98], as it will be further explained later.

Given the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, the goal is to estimate the nonlinear function f using LS-SVM, and to estimate the parameter vector $\boldsymbol{\beta}$ simultaneously. Using a regularized cost function, the proposed Partially Linear LS-SVM (PL-LSSVM) can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\beta},b,e_i} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^N e_i^2 \\ \text{s.t.} \quad & y_i = \boldsymbol{\beta}^T\boldsymbol{z}_i + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \qquad i = 1, \ldots, N, \end{aligned} \tag{4.2}$$

where $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ is the feature map. The solution is formalized in the following lemma.

Lemma 4.1. *Given a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, the solution to (4.2) is given by the system*

$$\left[\begin{array}{c|c|c} \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} & \mathbf{1} & \boldsymbol{Z} \\ \hline \mathbf{1}^T & 0 & \mathbf{0}_{1\times p} \\ \hline \boldsymbol{Z}^T & \mathbf{0}_{p\times 1} & \mathbf{0}_{p\times p} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \\ \hline \boldsymbol{\beta} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \\ \hline \mathbf{0}_{p\times 1} \end{array}\right], \tag{4.3}$$

where $\boldsymbol{y} = [y_1, \ldots, y_N]^T, \boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ *and* $\boldsymbol{Z} \in \mathbb{R}^{N \times p}$ *is the matrix of linear regressors* $\boldsymbol{z}_i$*. The solution is unique if* $\boldsymbol{Z}$ *has full column rank.*

Proof: Building the Lagrangian of the regularized cost function,

$$\mathscr{L}(\boldsymbol{w}, \boldsymbol{\beta}, b, e_i, \alpha_i) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 - \\ - \sum_{i=1}^{N} \alpha_i(\boldsymbol{\beta}^T\boldsymbol{z}_i + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i - y_i) \quad (4.4)$$

where $\alpha_i \in \mathbb{R}$ the Lagrange multipliers, and taking the optimality conditions, the following system of equations is obtained:

$$\begin{cases} \frac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0 & \rightarrow \quad \boldsymbol{w} = \sum_{j=1}^{N} \alpha_j \boldsymbol{\varphi}(x_j) \\ \frac{\partial \mathscr{L}}{\partial \boldsymbol{\beta}} = 0 & \rightarrow \quad \sum_{i=1}^{N} \alpha_i \boldsymbol{z}_i = \boldsymbol{0}_{p\times 1} \\ \frac{\partial \mathscr{L}}{\partial b} = 0 & \rightarrow \quad \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial \mathscr{L}}{\partial e_i} = 0 & \rightarrow \quad \alpha_i = \gamma e_i, \qquad i = 1, \ldots, N \\ \frac{\partial \mathscr{L}}{\partial \alpha_j} = 0 & \rightarrow \quad y_i = \boldsymbol{\beta}^T\boldsymbol{z}_i + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i, \quad i = 1, \ldots, N, \end{cases} \quad (4.5)$$

where $\boldsymbol{0}_{p\times 1}$ is a zero-valued vector of dimension $p \times 1$. With the application of Mercer's theorem [88] $\boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with a positive definite kernel K, it is possible to eliminate $\boldsymbol{w}$ and e_i, obtaining the final system (4.3) ■

The final model in dual form becomes

$$\hat{y}(\boldsymbol{x}, \boldsymbol{z}) = \hat{\boldsymbol{\beta}}^T\boldsymbol{z} + \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b. \quad (4.6)$$

The PL-LSSVM model defined by (4.3) always admits a unique solution for $(\boldsymbol{\alpha}, b, \hat{\boldsymbol{\beta}})$ if and only if both of the following conditions hold:

- $\boldsymbol{Z}$ has full column rank, and
- $\boldsymbol{Z}$ should not contain a column $c\boldsymbol{1}_N, c \in \mathbb{R}$.

In order to prove this, the following lemma is stated.

Lemma 4.2. *Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be a positive definite matrix; $\boldsymbol{B} \in \mathbb{R}^{N\times p}$; $\boldsymbol{d}_1, \boldsymbol{a}_1 \in \mathbb{R}^N$, and $\boldsymbol{d}_2, \boldsymbol{a}_2 \in \mathbb{R}^p$. The linear system defined by*

$$\left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{B}^T & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{d}_1 \\ \hline \boldsymbol{d}_2 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{array}\right], \tag{4.7}$$

has a unique solution if and only if $\boldsymbol{B}$ has full column rank.

Proof: The solutions for $\boldsymbol{d}_1, \boldsymbol{d}_2$ can be written as

$$\begin{aligned} \boldsymbol{d}_1 &= \boldsymbol{A}^{-1}\boldsymbol{a}_1 - \boldsymbol{A}^{-1}(\boldsymbol{B}^T\boldsymbol{A}\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{a}_1 - \boldsymbol{a}_2) \\ \boldsymbol{d}_2 &= (\boldsymbol{B}^T\boldsymbol{A}\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{a}_1 - \boldsymbol{a}_2). \end{aligned}$$

The unique solution exists if and only if the matrices $\boldsymbol{A}$ and $\boldsymbol{B}^T\boldsymbol{A}\boldsymbol{B}$ are invertible. As $\boldsymbol{A}$ is positive definite, it is always invertible. If $\boldsymbol{A}$ is positive definite, then $(\boldsymbol{B}^T\boldsymbol{A}\boldsymbol{B})$ is also positive definite, and therefore invertible, if and only if $\boldsymbol{B}$ has full column rank ([62], Observation 7.1.6, p.399). ■

In the case of standard LS-SVM (2.14), the matrix $\boldsymbol{A} = \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I}$ is positive definite, and the matrix $\boldsymbol{B}$ corresponds to a vector of ones, having full rank. Therefore, a unique solution always exists. In the case of the PL-LSSVM, $\boldsymbol{B} = [\mathbf{1}, \boldsymbol{Z}]$. By Lemma 4.2 a unique solution exists only if $\boldsymbol{B}$ has full rank, therefore $\boldsymbol{Z}$ needs to have full rank. As the first column in $\boldsymbol{B}$ is a vector of ones, it is also required that no such column (up to a constant) is found within $\boldsymbol{Z}$, otherwise there would be 2 linearly dependent columns in $\boldsymbol{B}$.

4.2 Links with traditional statistical techniques

Partially linear models of the form (4.1) have been used in many applications, starting from the seminal study of Engle *et al.* on the relation between electricity prices and temperature [25]. Statistical inference on the estimated parameters has been developed based on asymptotic theory and consistency results from nonparametric estimation theory [54]. Within the statistical literature, the model (4.1) is estimated by approximating f by a local smoother and solving a set of normal equations [107]

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T(I - S)\boldsymbol{Z})^{-1}\boldsymbol{Z}^T(I - S)\boldsymbol{y}, \tag{4.8}$$

where S is a smoother matrix. Usually S is related to local splines, or variants of the Nadaraya-Watson estimator [90]. In practice, usually $\boldsymbol{x}$ is constrained to have a very low dimensionality (typically one-dimensional).

Within the LS-SVM framework, by working with the equations from system (4.3), and assuming the data have been centered, it is possible to write

$$\boldsymbol{y} = \boldsymbol{Z\beta} + \boldsymbol{\Omega}[(\boldsymbol{\Omega} + \gamma^{-1}\mathbf{I})^{-\mathbf{1}}(\boldsymbol{y} - \boldsymbol{Z\beta})] + \boldsymbol{e}. \tag{4.9}$$

Pre-multiplying by $\boldsymbol{Z}^T$, and noting that $\boldsymbol{Z}^T\boldsymbol{e} = \boldsymbol{Z}^T\boldsymbol{\alpha}/\gamma = \mathbf{0}$ as given by one of the optimality conditions, yields

$$\boldsymbol{Z}^T\boldsymbol{y} = \boldsymbol{Z}^T S\boldsymbol{y} - \boldsymbol{Z}^T S\boldsymbol{Z\beta} + \boldsymbol{Z}^T\boldsymbol{Z\beta}, \tag{4.10}$$

where

$$S = \boldsymbol{\Omega}(\boldsymbol{\Omega} + \gamma^{-1}\mathbf{I})^{-\mathbf{1}}, \tag{4.11}$$

is the equivalent smoothing matrix obtained under the LS-SVM estimator. After solving for $\boldsymbol{\beta}$ in (4.10), one obtains (4.8), showing that the PL-LSSVM estimate of $\boldsymbol{\beta}$ is linked to the traditional statistical techniques by using the smoother S defined by (4.11). Moreover, the use of the LS-SVM improves compared to traditional local techniques as it can use a more general set of regressors in $\boldsymbol{x}$, regardless of its dimensionality. A unique solution is obtained for the global model, and the nonlinear behavior of f can be correctly identified using the kernel trick on the variables $\boldsymbol{x}$. Additionally, non-local basis functions can be used for the approximation of the nonlinear function f, for example, using a polynomial kernel.

Remark 4.1. [Unique representation of the linear part] *The linear parameter vector $\boldsymbol{\beta}$ in (4.2) is not uniquely defined if there exists a mapping $g : \mathbb{R}^n \to \mathbb{R}^p$ such that $g(\boldsymbol{x}) = \boldsymbol{z}$. This problem is already noted in [98]. It implies, for instance, that if a model contains only $\boldsymbol{x}$ variables, as $y_i = \boldsymbol{\beta}^T\boldsymbol{x}_i + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i$, then the parameter of the linear part $\boldsymbol{\beta}$ is not uniquely defined. To see this, we can write*

$$\begin{aligned} y_i &= \boldsymbol{\beta}^T\boldsymbol{x}_i + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i \\ &= \boldsymbol{\delta}^T\boldsymbol{x}_i + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) - \boldsymbol{\delta}^T\boldsymbol{x}_i + \boldsymbol{\beta}^T\boldsymbol{x}_i + b + e_i \\ &= \boldsymbol{\delta}^T\boldsymbol{x}_i + \tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{\varphi}}(\boldsymbol{x}_i) + b + e_i, \end{aligned}$$

$\forall \boldsymbol{\delta} \in \mathbb{R}^p$. From the last equation we can define equivalently a new nonlinear component

$$\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{\varphi}}(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) - \boldsymbol{\delta}^T\boldsymbol{x}_i + \boldsymbol{\beta}^T\boldsymbol{x}_i,$$

where a linear part can be captured by the new nonlinearity defined by $\tilde{\boldsymbol{w}} = [\boldsymbol{w}; -\boldsymbol{\delta} + \boldsymbol{\beta}], \tilde{\boldsymbol{\varphi}}(\boldsymbol{x}_i) = [\boldsymbol{\varphi}(\boldsymbol{x}_i); \boldsymbol{x}_i]$. As the function f in (4.1) is defined in a general way, it can approximate a general class of functions,

obviously including linear functions of the same inputs. The same reasoning can be applied to a function g such that $g(\boldsymbol{x}) = \boldsymbol{z}$, leading to the same identifiability problem. In practice, the partially linear model is used to estimate a linear response over certain variables when it is suspected that the total response also depends nonlinearly over a different set of variables, therefore the identifiability problem rarely happens in applied work.

4.3 Examples

In this section, some examples of the PL-LSSVM performance are shown. Its ability to identify correctly the linear and nonlinear components for some examples is assessed by Monte Carlo simulations. Its out-of-sample forecasting performance is examined for 3 model examples. In all cases, an RBF kernel is used and the parameters σ and γ are found by 10-fold cross validation over the corresponding training sample.

4.3.1 Methodology

The test cases are defined as follows:

- **Case I: Linear trend + static nonlinearity.** The model to be estimated is of the form $y_t = a_1 t + 2\text{sinc}(x_t) + e_t$, where the true value is $a_1 = 1.5$ and x_t is drawn from a uniform distribution over [0,2.5]; e_t is a Gaussian white noise of variance 0.02.

- **Case II: Static linearity + static nonlinearity.** The model to be estimated is of the form $y_t = a_1 z_t + 2\text{sinc}(x_t) + e_t$, where the true value is $a_1 = 1.5$; z_t and x_t are drawn from a uniform distribution over [0,2.5] and [0,1.5], respectively; e_t is Gaussian white noise of variance 0.02.

- **Case III: Linear autoregression + static nonlinearity** The model to be estimated is of the form $y_t = a_1 y_{t-1} + a_2 y_{t-2} + 2\text{sinc}(x_t) + e_t$, where the true value are $a_1 = 0.6, a_2 = 0.3$; the x_t is drawn from a normal distribution with zero mean and variance 5; e_t is Gaussian white noise of variance 0.02. This corresponds to a simple Hammerstein system.

- **Case IV: Autoregression with linear and nonlinear components.** The model to be estimated is of the form $y_t = a_1 y_{t-1} + a_2 y_{t-2} + \mathrm{sinc}(y_{t-3}) + e_t$, where the true values are $a_1 = 0.6, a_2 = 0.3$; e_t is Gaussian white noise of variance 0.02.

It worth noting that although the regressors contained in the linear part might be correlated with those under the nonlinear part, they are neither identical nor perfectly related. Therefore, there are no identifiability problems.

Identification of the linear and nonlinear components: Monte Carlo simulations are performed for all cases defined above. In order to compare the PL-LSSVM model with traditional techniques, Ordinary Least Squares (OLS) regression using all variables (in linear form) is implemented, as well as the partially linear model with the Nadaraya-Watson (NW) [90] smoother as in [107]. Data are generated by sampling the respective distributions and/or using the autoregressive forms where it is appropriate. For all cases the number of datapoints is $N = 200$, and the number of Monte Carlo repetitions is 1,000.

4.3.2 Results

Table 4.1 shows the results, as averages and standard deviations of the estimated parameters over 1,000 repetitions, together with the 10-fold crossvalidation mean squared error (CV-MSE). In the simple cases I and II, all techniques give similar performance for the identification of the linear parameters. For Case III, a bias is present in the OLS-based estimation of the linear parameter, due to the time-series nature of the problem. In Case IV both the NW and OLS show an important bias in each one of the estimates. The empirical distributions of the estimates obtained with this sampling is visualized in Figures 4.1 and 4.2, for the comparison between the estimated parameter $\hat{a}_1$ using PL-LSSVM (solid line) and NW (dashed line) for Cases III and IV, respectively. Although the general conditions for asymptotic consistence for the NW partially linear model estimator have been studied extensively, in practice it is not straightforward to verify if they are fulfilled by the problem at hand. By using Monte Carlo simulations for particular types of problems, it is possible to verify the properties of each estimator, especially when temporal or serial correlation is present in the data [51]. Regarding identification of the nonlinear part, Figure 4.3 shows

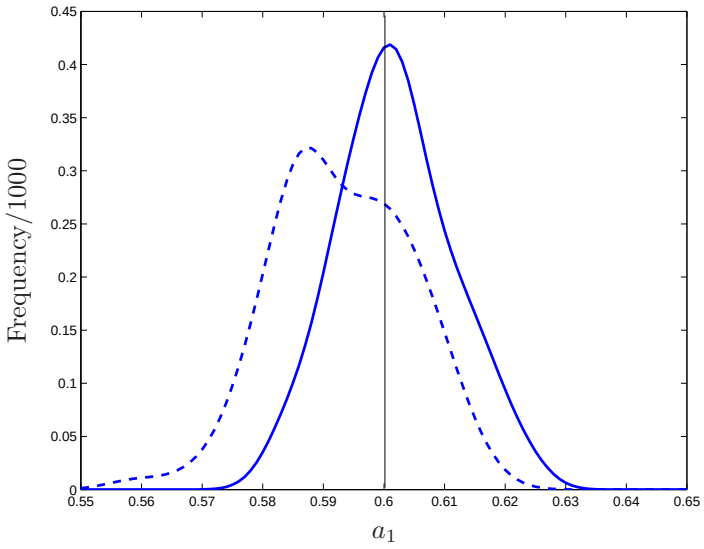

Figure 4.1: Empirical Distributions of the estimated parameter $\hat{a}_1$ under Case III using PL-LSSVM (solid line) and the Nadaraya-Watson (dashed line) estimator over 1,000 repetitions. The vertical line shows the 'true' value.

the identified nonlinear component of Case III. The 'o' shows the estimated nonlinear component of the model, given by $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b$, and the line shows the true value of the nonlinear function, with excellent performance. From the above examples it is clear that the PL-LSSVM estimator gives a satisfactory global accuracy, and at the same time it identifies successfully the linear part of each example.

4.4 Conclusions

Starting from the definition of LS-SVMs, we have shown how to define a feasible estimator for a partially linear model by extending the regression formulation in order to include a parametric part. The solution is shown to be unique and to exist under the usual requirements for a set of linear parametric regressors. This Partially Linear LS-SVM formulation is optimal in a least-squares sense, and allows to identify a general class of model structures. Its parametric part has the same structural form as classical

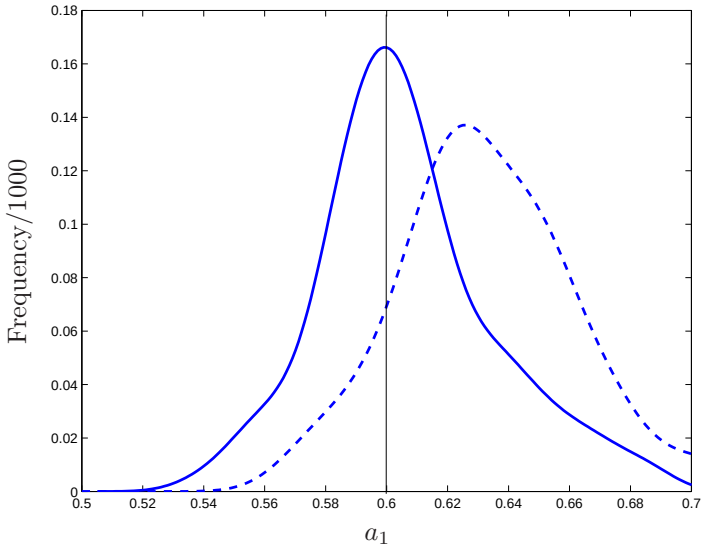

Figure 4.2: *Empirical Distributions of the estimated parameter $\hat{a}_1$ under Case IV using PL-LSSVM (solid line) and the Nadaraya-Watson estimator (dashed line) over 1,000 repetitions. The vertical line shows the 'true' value.*

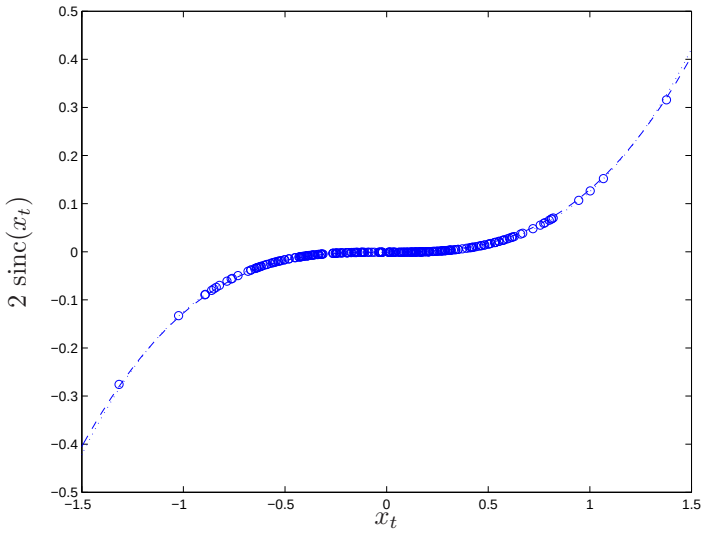

Figure 4.3: *The nonlinear part for the model of Case III, as estimated by the model ('o') and the true nonlinearity (line).*

	Estimates				**CV-MSE**	
	$\hat{a}_1$	$\sigma_{\hat{a}_1}$	$\hat{a}_2$	$\sigma_{\hat{a}_2}$	Mean	S.Dev
Case I						
PL-LSSVM	1.500	0.001	-	-	0.007	0.001
NW	1.500	0.003	-	-	0.09	0.01
OLS	1.498	0.007	-	-	0.19	0.02
Case II						
PL-LSSVM	1.50	0.01	-	-	0.008	0.001
NW	1.50	0.04	-	-	0.11	0.01
OLS	1.50	0.08	-	-	0.21	0.02
Case III						
PL-LSSVM	0.60	0.01	0.30	0.01	0.009	0.001
NW	0.59	0.01	0.30	0.01	0.17	0.01
OLS	0.57	0.01	0.32	0.01	0.25	0.01
Case IV						
PL-LSSVM	0.60	0.03	0.30	0.04	0.006	0.001
NW	0.63	0.03	0.26	0.04	0.07	0.01
OLS	1.16	0.05	-0.5	0.06	0.28	0.04

Table 4.1: *Mean and standard deviation for the parameter estimates and the CV-MSE over 1,000 repetitions.*

statistical methods, and it extends the classical notion of semiparametric regression by allowing explicitly to include any potential nonlinear regressor as the dimensionality of the system is defined in terms of the kernel matrix under Mercer's theorem. Practical examples over 4 particular types of models show the overall ability of the PL-LSSVM to identify the linear and nonlinear parts. Using Monte Carlo methods over 1,000 repetitions, it is clear that this method has a better global accuracy for the models and a better identification performance when compared to traditional techniques.

Chapter 5

LS-SVM with Autocorrelated Residuals

This chapter presents another contribution of this work. Typically it is assumed that the residuals on the LS-SVM regression are independent and identically distributed (i.i.d.). However, there are cases in which this assumption does not hold. When neglected, the presence of correlation in the error sequence can lead to severe problems not only in the identification of the function under study, but also in the predictions. In the nonparametric regression literature, it has been noted [6] that the presence of correlation in the error terms can mislead the identification of the nonlinear function when using a black-box identification technique. In plain terms, the black-box technique "learns" the structure in the nonlinear function together with the correlation structure in the errors. This problem can be solved by incorporating the knowledge of the correlation structure into the modeling stage. In this chapter, starting from prior knowledge on the correlation structure, we extend the LS-SVM regression formulation to incorporate autocorrelated residuals. We show that the solution embeds the correlation information into the kernel level for the approximation of the nonlinear function, and that the model structure leads to a predictor which also incorporates the correlation structure. By considering the correlation parameters as tuning hyperparameters, the least-squares

problem remains convex and Mercer's theorem can be applied. This chapter is organized as follows. Section 5.1 discusses the general model formulation and gives an introductory discussion. Section 5.2 discusses the derivations and the solution for the case of AR(1) errors. Section 5.3 extends the proposed formulation for the general AR(q) case. Section 5.4 shows examples where the inclusion of the prior knowledge on correlation substantially improves over the case where the correlation is neglected.

5.1 Regression Structure

The focus of this chapter is to estimate the nonlinear function f in the model

$$y_i = f(\boldsymbol{x}_i) + e_i \tag{5.1}$$

for the case where the sequence e_i is correlated.

The conditions under which the residual terms in (5.1) are correlated depend on the interpretation of the term e_i. The role of the e_i term has received different interpretations in the history of data modeling [96]. One interpretation says that the e_i term is considered a "random disturbance", as used in, for example, system identification [69, 76, 105]. Noise models are common in system identification, where it is assumed that the data generating process contains a component mainly driven by the inputs $\boldsymbol{x}$, and another component driven by noise. Under this interpretation, correlated noise appears when the system produces the output y_i not only as a result of the effect of the input $\boldsymbol{x}$ but also as a result of a hidden mechanism producing an observable random effect *correlated across observations*. For instance, they can be instrumentation errors or non-avoidable characteristics of the experimental setting (for example, a chemical plant in full operation).

A slightly different interpretation, that comes from the empirical point of view, gives the e_i term the meaning of "whatever is not explained by the information contained in $\boldsymbol{x}$". This is motivated by the practical rule of examining the whiteness of the residuals of a regression [76, 130], in such a way that if there is some correlation structure in the residuals, it means that there is still some information not captured by the variables in $\boldsymbol{x}$. This correlation can be caused by an error in the regression specification, which can be a missing variable, or a wrong functional form.

Consider as an example the case of the Boston housing dataset [24,56], where the goal was to study the effect of air pollution on housing values. The data consist of samples of median home values, with attributes such as nitrogen oxide concentration, crime rate, average number of rooms, percentage of nonretail business, and others. The goal is to build a regression that can predict the price of a house (y) from its attributes ($\boldsymbol{x}$). Consider the hypothetical case where the value of the house also depends on the distance from the local police station. Consider the situation where a house that is closer to the police station has a higher value because of security issues. If the attribute "distance to the police station" is not contained in the set of observed attributes $\boldsymbol{x}$ that the researcher is using to estimate the regression, the residuals will show a serial correlation between neighboring houses. There is something that is not completely explained by the variables in $\boldsymbol{x}$ in such a way that it appears as a correlation structure in the residuals. On the other hand, a particular variable may affect the price of the house in a nonlinear way, producing a sequence of correlated residuals if the regression specification neglects the nonlinear effect. In this case, from the practical point of view, the researcher has 2 options: either keep looking for the perfect set of variables to be included in $\boldsymbol{x}$, which may be expensive, or simply unfeasible; or otherwise correct the effect of the correlation present in the residuals. Moreover, the goal of a modeling task can be formulated as to find the best possible representation of the system *for a given set of inputs* $\boldsymbol{x}$. From this point of view, very usual in empirical practice, the best the researcher can do is to improve the correlation structure of the residuals in order to estimate correctly the effect of the $\boldsymbol{x}$ variables. This has been typically the case for datasets related to social, economic or medical studies, where the goal is to build a representation based on a limited set of variables [5, 104], and there is autocorrelation in the residuals which has to be corrected [79].

It is important to notice that the (seemingly) static regression problem between y and $\boldsymbol{x}$ (5.1) gets a new dimension given by the correlation of the residuals. In the case of autocorrelated residuals, there is a new time dimension embedded in the sequence of residuals, in such a way that consecutive residuals are correlated.

In this chapter, the focus is towards time series prediction rather than cross-sectional analysis. Therefore, it has been assumed that the correlation structure in the residuals is given by autocorrelation (and not serial

correlation). Consider the regression structure,

$$\begin{cases} y_i = f(\boldsymbol{x}_i) + e_i \\ A(z^{-1})e_i = r_i \end{cases} \tag{5.2}$$

for $i = 2, \ldots, N$. The residuals e_i of the first equation are uncorrelated with the input vector $\boldsymbol{x}_i$, and the sequence e_i is assumed to follow an invertible AR(q) process described by

$$A(z^{-1})e_i = r_i \tag{5.3}$$

where r_i is a white noise sequence with zero mean and constant variance σ_u^2, and where $A(z^{-1})$ is a monic polynomial in the lag operator z^{-1} with unknown parameters $a_j, j = 1, \ldots, q$,

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_q z^{-q}. \tag{5.4}$$

with $z^{-1}e_i = e_{i-1}$. This leads to the equivalent representation of (5.2),

$$\begin{cases} y_i = f(\boldsymbol{x}_i) + e_i \\ e_i + a_1 e_{i-1} + a_2 e_{i-2} + \ldots + a_q e_{i-q} = r_i \end{cases} \tag{5.5}$$

Prior knowledge on the existence and AR(q) structure of the correlation is assumed. Therefore, the problem of detecting correlation is not addressed. At the same time, the AR(q) parameters $a_j, j = 1 \ldots, q$, are considered as tuning parameters rather than to be optimized at the training sample. As a result, the problem remains convex, as it will be further explained.

5.2 LS-SVM with AR(1) errors

For clarity of the presentation, the derivations are first presented for the case of $q = 1$, for which $A(z^{-1})e_i = e_i - \rho e_{i-1}$. This case is often used in applied work (for example, in the seminal work on electricity prices by the Nobel laureates Engle and Granger [25]), and provides an interesting starting point for further analysis. The inclusion of AR(1) errors to the LS-SVM regression can be formulated as follows. Given the sample of N points $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ and the model structure (5.5), the following optimization problem with a regularized cost function is formulated:

$$\min_{\boldsymbol{w},b,r_i,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 \tag{5.6}$$

$$\text{s.t.} \quad \begin{cases} y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) + b + e_i \\ e_i = \rho e_{i-1} + r_i \end{cases}$$

for $i = 2, \ldots, N$, where γ is a regularization constant and the AR(1) coefficient ρ is a tuning parameter satisfying $|\rho| < 1$ (invertibility condition of the AR(1) process). The nonlinear function f from (5.2) is parameterized as $f(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) + b$, where the nonlinear function $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ is the mapping to a high dimensional (and possibly infinite dimensional) feature space. By eliminating e_i, the following equivalent problem is obtained:

$$\min_{\boldsymbol{w}, b, r_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma \frac{1}{2} \sum_{i=q+1}^{N} r_i^2 \tag{5.7}$$

$$\text{s.t. } y_i = \rho y_{i-1} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) - \rho \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_{i-1}) + b(1-\rho) + r_i,$$

for $i = 2, \ldots, N$, corresponding to the case of standard LS-SVM regression for nonlinear identification of a *dynamical* regression structure, where a time-series character is introduced into the model by the correlation of the residuals e_i. The residuals r_i of this new model (5.7) are uncorrelated by construction and, therefore, standard LS-SVM regression can be applied. The solution is formalized in the following lemma.

Lemma 5.1. *Given a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, with $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\varphi}(\boldsymbol{x}_i)^T \boldsymbol{\varphi}(\boldsymbol{x}_j)$, the solution of (5.7) is given by the dual problem*

$$\left[\begin{array}{c|c} 0 & \mathbf{1}^T \\ \hline \mathbf{1} & \boldsymbol{\Omega}^{(\rho)} + \gamma^{-1}\mathbf{I} \end{array}\right] \left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \tilde{\boldsymbol{y}} \end{array}\right], \tag{5.8}$$

with $\tilde{\boldsymbol{y}} = [y_2 - \rho y_1, \ldots, y_N - \rho y_{N-1}]^T, \boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{N-1}]^T$, and $\boldsymbol{\Omega}_{ij}^{(\rho)} = K(\boldsymbol{x}_{i+1}, \boldsymbol{x}_{j+1}) - \rho K(\boldsymbol{x}_i, \boldsymbol{x}_{j+1}) - \rho K(\boldsymbol{x}_{i+1}, \boldsymbol{x}_j) + \rho^2 K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ $\forall i, j = 1 \ldots, N-1$.

Proof: Consider the Lagrangian of problem (5.7)

$$\begin{aligned} \mathscr{L}(\boldsymbol{w}, b, r_i; \boldsymbol{\alpha}) \;=\; & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 - \sum_{i=2}^{N} \alpha_{i-1}[\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) \\ & -\rho\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_{i-1}) + \rho y_{i-1} - y_i - r_i], \end{aligned} \tag{5.9}$$

where $\alpha_i \in \mathbb{R}, i = 1, \ldots, N-1$ are the Lagrange multipliers. Taking the optimality conditions $\frac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0$, $\frac{\partial \mathscr{L}}{\partial b} = 0$, $\frac{\partial \mathscr{L}}{\partial r_i} = 0$, $\frac{\partial \mathscr{L}}{\partial \alpha_i} = 0$ yields

$$\boldsymbol{w} \;=\; \sum_{i=2}^{N} \alpha_{i-1}[\boldsymbol{\varphi}(\boldsymbol{x}_i) - \rho\boldsymbol{\varphi}(\boldsymbol{x}_{i-1})],$$

$$\begin{aligned} r_i &= \alpha_{i-1}/\gamma, \quad i = 2, \ldots, N, \\ 0 &= \sum_{i=1}^{N-1} \alpha_i, \\ y_i &= \rho y_{i-1} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_i) - \rho \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_{i-1}) \\ &\quad + b(1-\rho) + r_i, \quad i = 2, \ldots, N. \end{aligned}$$

With the application of Mercer's theorem [129] $\boldsymbol{\varphi}(\boldsymbol{x}_i)^T \boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with a positive definite kernel K, we can eliminate $\boldsymbol{w}$ and r_i, obtaining $y_i - \rho y_{i-1} = \sum_{i=2}^{N} \alpha_{i-1}(K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \rho K(\boldsymbol{x}_{i-1}, \boldsymbol{x}_j) - \rho K(\boldsymbol{x}_i, \boldsymbol{x}_{j-1}) + \rho^2 K(\boldsymbol{x}_{i-1}, \boldsymbol{x}_{j-1})) + b + \frac{\alpha_i}{\gamma}$. Building the kernel matrix $\boldsymbol{\Omega}_{ij}^{(\rho)}$ and writing the equations in matrix notation gives the final system (5.8) ■

Remark 5.1. [Equivalent Kernel] *The final approximation of f in the original model (5.2) with $q = 1$ can be expressed in dual space as*

$$\hat{f}(\boldsymbol{x}_i) = \sum_{j=2}^{N} \alpha_{j-1} K_{eq}(\boldsymbol{x}_j, \boldsymbol{x}_i) + b \tag{5.10}$$

where $K_{eq}(\boldsymbol{x}_j, \boldsymbol{x}_i) = K(\boldsymbol{x}_j, \boldsymbol{x}_i) - \rho K(\boldsymbol{x}_{j-1}, \boldsymbol{x}_i)$ is an equivalent kernel which embodies the information on the AR(1) error correlation.

Remark 5.2. [Partially Linear Structure] *The existence of correlated errors in (5.5) results into new dynamics into the system, leading to the model structure (5.7) which is a partially linear model [34, 107] with a very specific restriction on the coefficients: the past output y_{i-1} is included as a linear term with coefficient ρ, and the past input vector $\boldsymbol{x}_{i-1}$ is included under the nonlinear function which, in turn, is weighted by the value $-\rho$.*

Remark 5.3. [Considering ρ as an unknown] *If ρ is considered as an unknown instead of a tuning parameter in (5.6), an additional optimality condition from the Lagrangian $\frac{\partial \mathscr{L}}{\partial \rho} = 0$ gives*

$$\sum_{i=2}^{N} \alpha_{i-1}[y_{i-1} - \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_{i-1}) - b] = 0.$$

Noting that $e_{i-1} = y_{i-1} - \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}_{i-1}) - b$ and $\alpha_{i-1} = r_i \gamma = [e_i - \rho e_{i-1}]\gamma$, the estimate $\hat{\rho}$ is obtained as a solution of

$$\sum_{i=2}^{N} [e_i - \rho e_{i-1}] e_{i-1} = 0, \tag{5.11}$$

or,

$$\hat{\rho} = \frac{\sum_{i=2}^{N} e_i e_{i-1}}{\sum_{i=2}^{N} e_{i-1}^2}, \tag{5.12}$$

corresponding to the ordinary least squares (OLS) estimator of the slope parameter from a linear regression of e_i on e_{i-1}. This is a very intuitive result, but unfortunately the sequence e_i is unobserved. Moreover, considering ρ as an unknown parameter in (5.6) gives rise to a non-convex problem, as the remaining optimality conditions include products of ρ and the other unknowns. Thus, considering ρ as an unknown in (5.6), makes the optimization problem more difficult to solve.

Remark 5.4. [Considering ρ as a tuning parameter] *The parameter ρ is considered as a tuning parameter in order to work with a feasible convex optimization problem in which Mercer's Theorem can be applied and a unique solution can be obtained. Therefore, the parameter ρ is determined on another level (for example, by means of cross-validation) to yield a good generalization performance of the model, although this does not necessarily mean that the optimality condition (5.12), obtained for the case where ρ is an unknown in (5.6), is enforced. In this way, the selected ρ is the value that gives the best cross-validation performance. This approach may increase the computational load, as each time a grid of possible values has to be defined for ρ, which may become computationally intensive for a general AR(q) case with $q > 1$. However, the definition of possible values can be guided from theoretical ranges for allowed values of ρ, derived from the invertibility condition of the AR(q) process: for $q = 1$, we have $|\rho| < 1$; for $q = 2$, a sufficient condition is $|\rho_1 + \rho_2| < 1$. In general it is required for all roots of the equation $1 + a_1 x + a_2 x^2 + \ldots + a_q x^q = 0$ to be outside the unit circle [51].*

5.3 Autocorrelated Residuals: The general AR(q) case

Consider the general case for a model with a AR(q) noise process:

$$\begin{cases} y_i = f(\boldsymbol{x}_i) + e_i \\ A(z^{-1}) e_i = r_i \end{cases} \tag{5.13}$$

for $i = 2, \ldots, N$, and where

$$A(z^{-1}) e_i = r_i \tag{5.14}$$

is an invertible autoregressive process of order q. The following regression problem can be formulated with LS-SVM:

$$\min_{\boldsymbol{w},b,r_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=q+1}^{N} r_i^2 \tag{5.15}$$

$$\text{s.t.} \quad A(z^{-1})y_i = A(z^{-1})(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_i) + b) + r_i, \quad i = q+1,\ldots,N$$

The solution is formulated in the following lemma.

Lemma 5.2. *Given a positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, with $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j)$, the solution to (5.15) with $A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \ldots + a_qz^{-q} = \sum_{k=0}^{q} a_kz^{-k}$ is given by the dual problem*

$$\left[\begin{array}{c|c} 0 & \mathbf{1}^T \\ \hline \mathbf{1} & \boldsymbol{\Omega}^{(A)} + \gamma^{-1}\mathbf{I} \end{array}\right]\left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \tilde{\boldsymbol{y}} \end{array}\right], \tag{5.16}$$

where $\tilde{\boldsymbol{y}} = [A(z^{-1})y_{q+1},\ldots,A(z^{-1})y_N]^T, \boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_{N-q}]^T$, and $\boldsymbol{\Omega}^{(A)}$ is the kernel matrix with $\boldsymbol{\Omega}^{(A)}_{ij} = \sum_{k=0}^{q}\sum_{l=0}^{q} a_ka_lK(\boldsymbol{x}_{i+q-k}, \boldsymbol{x}_{j+q-l})$ $\forall i,j = 1\ldots,N-q$.

Proof: Consider the Lagrangian of problem (5.15)

$$\begin{aligned}\mathscr{L}(\boldsymbol{w},b,r_i;\boldsymbol{\alpha}) = {} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=q+1}^{N} r_i^2 \\ & - \sum_{i=q+1}^{N}\alpha_{i-q}[\boldsymbol{w}^TA(z^{-1})\boldsymbol{\varphi}(\boldsymbol{x}_i) - A(z^{-1})y_i + A(z^{-1})b - r_i],\end{aligned}$$

where $\alpha_i \in \mathbb{R}, i = 1,\ldots,N-q$ are the Lagrange multipliers. Taking the optimality conditions $\frac{\partial\mathscr{L}}{\partial\boldsymbol{w}} = 0$, $\frac{\partial\mathscr{L}}{\partial b} = 0$, $\frac{\partial\mathscr{L}}{\partial r_i} = 0$, $\frac{\partial\mathscr{L}}{\partial\alpha_{k-q}} = 0$ yields

$$\begin{aligned}\boldsymbol{w} &= \sum_{i=q+1}^{N}\alpha_{i-q}[A(z^{-1})\boldsymbol{\varphi}(\boldsymbol{x}_i)], \\ r_i &= \alpha_{i-q}/\gamma, \quad i = q+1,\ldots,N, \\ 0 &= \sum_{q=1}^{N-q}\alpha_i,\end{aligned}$$

$$A(z^{-1})y_i \quad = \quad \boldsymbol{w}^T A(z^{-1})\boldsymbol{\varphi}(\boldsymbol{x}) + A(z^{-1})b + r_i, \quad i = q+1, \ldots, N.$$

With the application of Mercer's theorem [129] $\boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with a positive definite kernel K, it is possible to eliminate $\boldsymbol{w}$ and r_i. Building the kernel matrix $\boldsymbol{\Omega}^{(A)}_{i,j}$ and writing the equations in matrix notation gives the final system (5.16) ■

Remark 5.5. [Equivalent Kernel and Final Predictor] *The equivalent kernel for the general AR(q) case is given by:*

$$K_{eq}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=0}^{q}\sum_{l=0}^{q} a_k a_l K(\boldsymbol{x}_{i+q-k}, \boldsymbol{x}_{j+q-l}). \tag{5.17}$$

The final model can be written in terms of $\tilde{y} = A(z^{-1})y$. Given a new datapoint $\boldsymbol{x}_{N+1}$, the predicted $\tilde{y}(\boldsymbol{x}_{N+1})$ is given as

$$\tilde{y}(\boldsymbol{x}_{N+1}) = \sum_{j=q+1}^{N}\sum_{k=0}^{q}\sum_{l=0}^{q} a_k a_l K(\boldsymbol{x}_{j+q-k}, \boldsymbol{x}_{N+1+q-l})\alpha_{j-q} + b\sum_{j=0}^{q} a_j, \tag{5.18}$$

from which the predicted $y(\boldsymbol{x}_{N+1})$ can be recovered from

$$y(\boldsymbol{x}_{N+1}) = \tilde{y}(\boldsymbol{x}_{N+1}) - \sum_{j=1}^{N} a_j y(\boldsymbol{x}_{N+1-j}). \tag{5.19}$$

The final summation in (5.19) makes it explicit that the new datapoints are indeed correlated with the previous points in the sample. As mentioned before, the seemingly static regression obtains the temporal dimension from the correlation of the residuals.

Remark 5.6. [Effect of the polynomial $A(z^{-1})$] *The polynomial $A(z^{-1})$ acts as a linear operator. It transforms the datapoint y_t into $\tilde{y}_t = y_t + a_1 y_{t-1} + a_2 y_{t-2} + a_q y_{t-q}$. In the econometric literature, this is commonly referred as "quasi-differencing" [51,67]. In the linear regression context, this linear operator commutes with the coefficient of the regression, thus making it possible to apply an iterative estimation process. In this process, first the data $\{y_t, \boldsymbol{x}_t\}$ is preprocessed by "quasi-differencing", and then the regression is estimated on the transformed data $\{\tilde{y}_t, \tilde{\boldsymbol{x}}_t\}$, i.e. $A(z^{-1})(\boldsymbol{w}^T\boldsymbol{x}_t) = \boldsymbol{w}^T A(z^{-1})\boldsymbol{x}_t = \boldsymbol{w}^t\tilde{\boldsymbol{x}}_t$. In the econometric literature, a practical iterative method for the case of AR(1) residuals has become very popular in applied work (so-called "Cochrane-Orcutt" method [10, 16]). In the case of a*

nonlinear regression with LS-SVM, this is not possible, because the linear operator does not commute with the nonlinear mapping. It is not true that $A(z^{-1})(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_t)) = \boldsymbol{w}^T\boldsymbol{\varphi}(A(z^{-1})\boldsymbol{x}_t) = \boldsymbol{w}^t\boldsymbol{\varphi}(\tilde{\boldsymbol{x}}_t)$*, and therefore it is not possible to transform the data first and then perform the nonlinear regression on the transformed data.*

5.4 Examples

In this section, two examples are considered to illustrate the effect of AR(1) residuals. The first is a static regression model, the second is an autoregressive formulation. In each case, an RBF kernel is used, and the hyperparameters are tuned by 10-fold cross-validation. By assumption, $|\rho| < 1$. The considered values for the tuning parameter ρ range from -0.9 to 0.9 with 0.1 steps. Each example involves the estimation of the correlation-corrected LS-SVM (C-LS-SVM) and standard LS-SVM for comparison.

Example 1: Static Nonlinearity.

Consider the following example where the true underlying system (5.2) is defined to contain a static formulation $f(x) = 1 - 6x + 36x^2 - 53x^3 + 22x^5$. The input values x_i are sampled i.i.d. from a uniform distribution between 0 and 1, with $N = 100$ datapoints. The error sequence e_i is built using $\rho = 0.7$ and $\sigma_u^2 = 0.5$. In this case, the original system is static, and the correlation induces a dynamical behavior in the observed values. Figure 5.1 shows the plot of y versus x, in order to visualize the true polynomial function as a function of x. The true f function is shown as a thin line, and the estimated function from (5.10) as a thick one. For comparison, the estimated function with standard LS-SVM (neglecting correlation) is shown in dashed-line. Clearly, the estimation with the corrected LS-SVM can better identify the true function, whereas the standard LS-SVM mixes the true function with the correlation structure. The parameter ρ minimizing the cross-validation mean squared error (MSE) coincides with the true AR(1) parameter 0.7. This example of a static nonlinearity already shows the effect of the error correlation, where the apparently independent sequence of inputs and outputs obtains a temporal correlation by means of the residuals of the equation.

Example 2: Autoregressive model

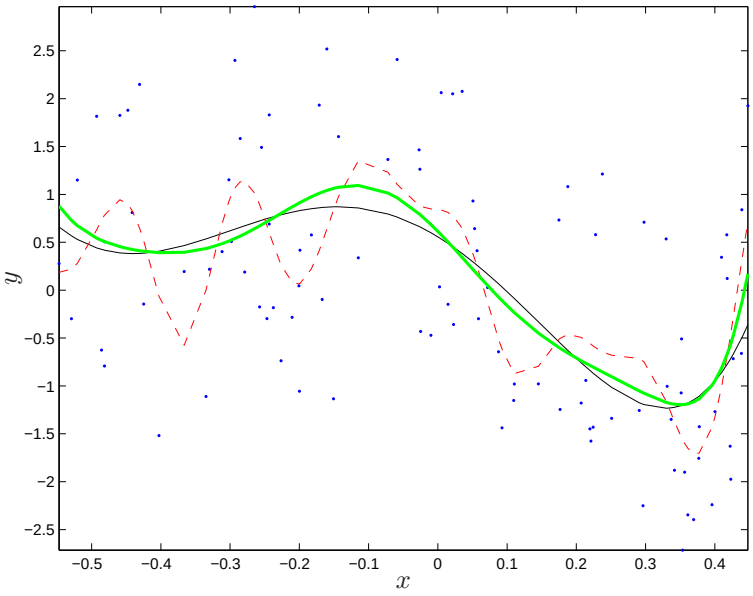

Figure 5.1: *True (thin) function and identified functions estimated with AR(1)-LSSVM (thick) and standard LS-SVM (dashed) for a static nonlinear function given by* $f(x) = 1 - 6x + 36x^2 - 53x^3 + 22x^5$.

This example considers the identification of the model

$$\begin{cases} y_i = 2 \cdot \mathrm{sinc}(y_{i-1}) + e_i \\ e_i - \rho e_{i-1} = r_i \end{cases} \tag{5.20}$$

generated with $\rho = 0.6$, $\sigma_u = 0.1$ for 150 datapoints. The first 100 points are used for identification, and the remaining 50 points for out-of-sample assessment of the prediction performance.

- *Identification of the AR(1) parameter.* Following the standard methodology, 10-fold cross-validation is performed to select the hyperparameters γ (regularization term), σ (RBF kernel parameter) and the ρ (the AR(1) parameter). Figure 5.2 (top) shows the cross-validation MSE for different combinations of hyperparameters, plotted for the values of ρ. In other words, for a given ρ, different MSE results are obtained depending on the combinations of σ and γ. The best performance is obtained for ρ=0.6 corresponding to the true value of the AR(1) process.

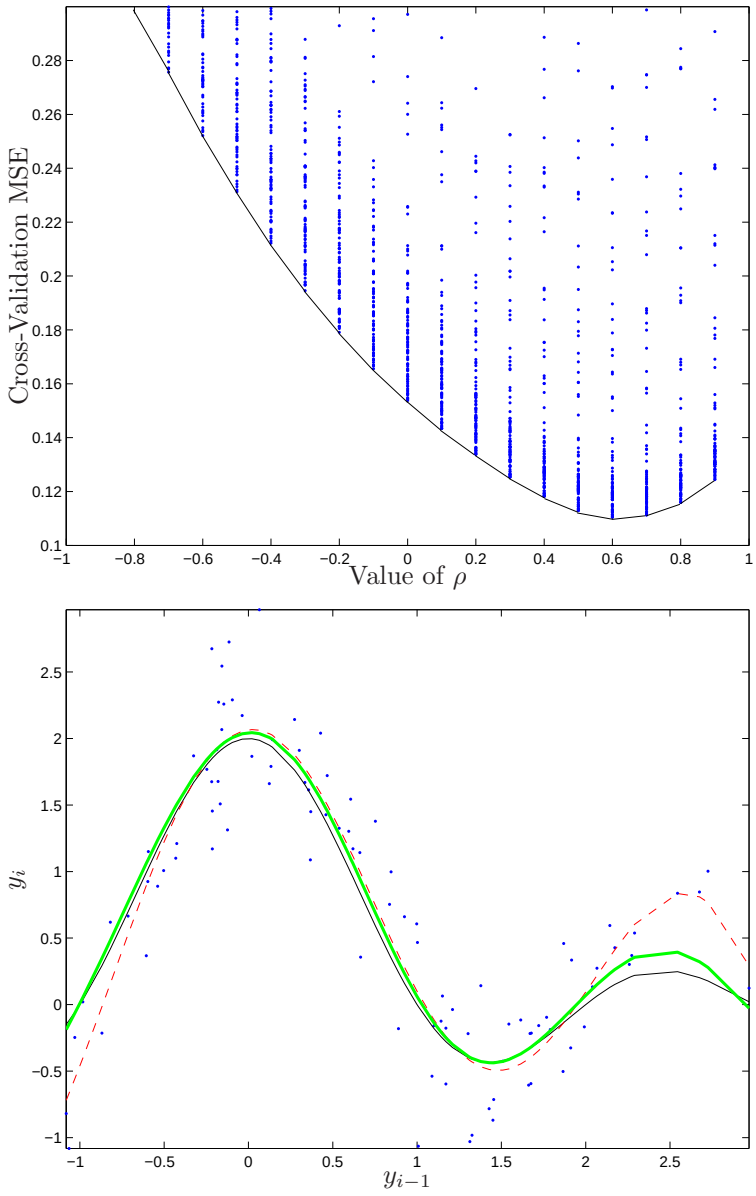

Figure 5.2: *(Top) Evolution of the cross-validation MSE for different combination of hyperparameters. The optimal performance is found at ρ=0.6.(Bottom) True (thin) function and the identified functions estimated with AR(1)-LSSVM (thick) and standard LS-SVM (dashed) for Example 2.*

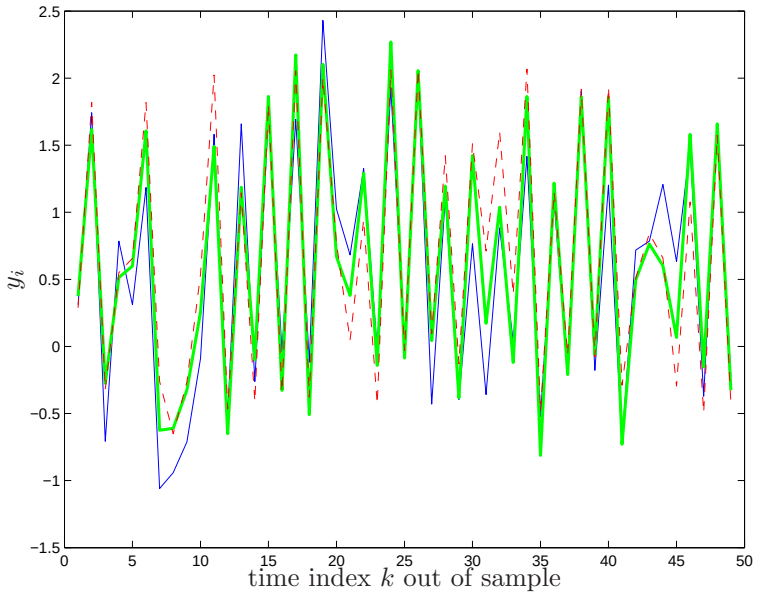

Figure 5.3: *Out-of-sample predictions obtained with C-LSSVM (thick) and standard LS-SVM (dashed) compared to the actual values (thin line) for Example 2.*

- *Identification of the nonlinear function.* Once the hyperparameters are selected, the approximation of f is obtained from (5.10). Figure 5.2 (bottom) shows the training points (dots), the identified function $\hat{f}$ (thick line), the true function (thin line) and the approximation obtained with standard LS-SVM (dashed line) for comparison. As in the previous example, the corrected LS-SVM is able to separate the correlation effects from the nonlinear function.

- *Prediction Performance.* Using the expression (5.19), out-of-sample predictions are computed for the system (5.20) for the next 50 datapoints. Table 5.1 shows the MSE calculated over the test set, compared to the results obtained from predicting using standard LS-SVM. The better performance of the correlation-corrected LS-SVM reflects the fact that the optimal predictor includes all information on the model structure, whereas the standard LS-SVM considers all dynamical effects being due to the nonlinear function only. Figure 5.3 shows the actual values (thin) and the predictions generated by

C-LSSVM (thick) and standard LS-SVM (dashed).

Performance	LS-SVM	C-LS-SVM
MSE in-sample	0.13	0.09
MSE cross-validation	0.17	0.10
MSE out-of-sample	0.18	0.09

Table 5.1: *In-sample, cross-validation and out-of-sample performance of the models for Example 2.*

5.5 Conclusions

In this chapter the LS-SVM formulation for regression is extended to incorporate autocorrelated residuals. Starting from the prior knowledge of the correlation structure, the modeling is treated as a convex problem where the coefficients of the AR residual process are considered to be tuning parameters. The dual solution of the model incorporates the correlation information into the kernel level. Additionally, the optimal one-step-ahead predictor includes the correlation structure explicitly. The correlation structure causes a very specific dynamic behavior in the final model. Practical examples show how the inclusion of the correlation structure into the model gives a much better identification of the nonlinear function, and better out-of-sample performance in terms of prediction and simulation.

Part II

Nonlinear System Identification

Chapter 6

Nonlinear System Identification with LS-SVM

This chapter shifts the estimation techniques developed in Part I into the context of nonlinear system identification, with the contribution of providing a general framework in which a nonlinear model structure can be defined following a modular approach. In the context of system identification, the problem of finding a model structure from available data has been studied extensively for the case of linear models [3, 76, 108, 111]. In the context of nonlinear system identification, however, the task of model selection and estimation is more complicated. One reason is that there is a vast choice of nonlinear estimation techniques that can be used for this purpose, giving rise to a very rich spectrum of options available for the user. Neural networks [57], wavelets [141], nonparametric kernel regression [52], and others, confront the user with the challenge of not only defining the model structure and obtaining an estimation from available data, but also to deal with the intrinsic architecture of the selected nonlinear technique, that can have its advantages, limitations and drawbacks. For the case of neural networks, for instance, it is also required to be familiar with different training algorithms, like e.g. backpropagation, early stopping criterion, being aware of multiple local minima, etc. The estimation techniques developed in the first part of this work, on

the other hand, can be used for nonlinear system identification following a modular approach built around a convex optimization problem with a unique solution. This provides an important degree of flexibility in the design of the model structure; the estimator is based on LS-SVMs, but it contains different elements that can be tailored to the prior knowledge of the problem at hand, following the rule of "do not estimate what you already know" [105]. Moreover, thanks to this flexibility, the path from a full linear parameterization towards a nonlinear specification can be made more gradually, which is important for practical applications. In this work 2 model structures are considered: a NARX model, being a Nonlinear AutoRegression with eXogenous inputs [69, 105], and an AR-NARX model, that is, a Nonlinear AutoRegression with eXogenous inputs and AutoRegressive residuals [48]. Each model structure can be formulated in terms of the different parameterizations using the results from Part I. This chapter is structured as follows. Section 6.1 describes the model structures. Section 6.2 describes the different nonlinear parameterizations for each of the model structures. The estimation methodology in dual space and primal space are given in Section 6.3 and Section 6.4, respectively. Section 6.5 shows illustrative examples.

6.1 Model Structures

Consider the following regression vector $\boldsymbol{z}_t \in \mathbb{R}^n$ $\boldsymbol{z}_t = [y_{t-1}; \cdots; y_{t-p}; \boldsymbol{u}_t; \boldsymbol{u}_{t-1}; \cdots; \boldsymbol{u}_{t-q}]$ containing p past values of the output $y_t \in \mathbb{R}$ and q past values of input vectors $\boldsymbol{u}_t \in \mathbb{R}^{N_u}$. A NARX model structure [69, 76, 105] can be formulated as

$$y_t = g(\boldsymbol{z}_t) + e_t \tag{6.1}$$

where the error term e_t is assumed to be i.i.d. with zero mean and constant variance σ_e^2. The AR-NARX [38, 48, 76] model structure incorporates an autoregressive process on the error terms e_t. The AR(1)-NARX model structure can be described as

$$\begin{cases} y_t = g(\boldsymbol{z}_t) + e_t \\ A(z^{-1})e_t = r_t \end{cases} \tag{6.2}$$

The residuals e_t of the first equation are uncorrelated with the input vector $\boldsymbol{z}_t$, and the sequence e_t is assumed to follow an invertible autoregressive AR(q) process described by

$$A(z^{-1})e_t = r_t \tag{6.3}$$

where r_t is a white noise sequence with zero mean and constant variance σ_u^2, and where $A(z^{-1})$ is a monic polynomial in the lag operator z^{-1} with unknown parameters $a_j, j = 1, \ldots, q$,

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_q z^{-q}. \tag{6.4}$$

with $z^{-1}e_i = e_{i-1}$. For clarity purposes (and practical considerations, taking into account the tuning of the parameters of the polynomial $A(z^{-1})$), only the AR(1) case is considered in the remaining of this chapter.

6.2 Model Parameterizations

For the parameterization of the function $g(\cdot)$ in (6.1) or (6.2) the following alternatives are considered.

6.2.1 Black-Box Parameterization

The nonlinear function $g(\cdot)$ for a NARX (6.1) or AR-NARX (6.2) structure is parameterized under a black-box formulation in primal space using LS-SVMs (2.13):

$$g(\boldsymbol{z}_t) = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_t) + b \tag{6.5}$$

where b is a constant (bias) term, and $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{N_h}$ is the feature map from the input space to the so-called feature space (of dimension N_h which can be possibly infinite). As explained in previous chapters, this feature map is used in relation to a Mercer kernel [114, 129], in such a way that the feature map is not computed explicitly.

6.2.2 Partially Linear Parameterization

In this case, some of the regressors are included as linear terms, and others are included under a nonlinear black-box term. Consider a partition of the

regression vector $\boldsymbol{z}_t$ as follows. Consider the set

$$\mathscr{Z} = \{x : x \text{ is a component of the vector } \boldsymbol{z}_t\},$$

and define an arbitrary partition

$$\mathscr{Z} = \mathscr{Z}_A \cup \mathscr{Z}_B$$

with

$$\mathscr{Z}_A \cap \mathscr{Z}_B = \emptyset. \tag{6.6}$$

Define a vector $\boldsymbol{z}_{A,t} \in \mathbb{R}^{N_a}$ with regressors $\boldsymbol{x} \in \mathscr{Z}_A$, and a vector $\boldsymbol{z}_{B,t} \in \mathbb{R}^{N_b}$ with regressors $\boldsymbol{x} \in \mathscr{Z}_B$. The original regression vector is thus partitioned as $\boldsymbol{z}_t = [\boldsymbol{z}_{A,t}\ ,\ \boldsymbol{z}_{B,t}]$. The subscript A (resp. B) represents the subset of regressors entering linearly (resp. nonlinearly) into the model. The nonlinear component of this Partially Linear parameterization is expressed under a black-box formulation using LS-SVMs. The nonlinear function $g(\cdot)$ for a PL-NARX (6.1) or a PL-AR-NARX (6.2) is parameterized as

$$g(\boldsymbol{z}_t) = \boldsymbol{\beta}^T \boldsymbol{z}_{A,t} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,t}) + b \tag{6.7}$$

for a given partition $\boldsymbol{z}_t = [\boldsymbol{z}_{A,t}\ ,\ \boldsymbol{z}_{B,t}]$. The condition (6.6) is imposed to ensure a unique representation of the parameter $\boldsymbol{\beta}$, as discussed on remark 4.1.

6.3 Model Estimation in Dual Space

The different nonlinear model structures can be estimated using the LS-SVM regression framework. Starting from a given dataset $\{\boldsymbol{z}_i, y_i\}_{i=1}^N$, the different estimation problems are presented for each of the model structures defined in section 6.2.

6.3.1 Black-Box NARX Model

For the NARX model (6.1), with $g(\cdot)$ parameterized as in (6.5), the following optimization problem with a regularized cost function is formulated:

$$\min_{\boldsymbol{w},b,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \tag{6.8}$$

$$\text{s.t.} \quad y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_i) + b + e_i, \quad i = 1, \ldots, N.$$

where γ is a regularization constant. The solution is obtained using (2.15) from lemma 2.1.

6.3.2 PL-NARX Model: Considering a Partially Linear Structure

For the PL-NARX model (6.2), with $g(\cdot)$ parameterized as in (6.7), the following optimization problem with a regularized cost function is formulated:

$$\min_{\boldsymbol{w},b,e_i,\boldsymbol{\beta}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \tag{6.9}$$

$$\text{s.t.} \quad y_i = \boldsymbol{\beta}^T \boldsymbol{z}_{A,i} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,i}) + b + e_i, \quad i = 1, \ldots, N,$$

where γ is a regularization constant. The solution is obtained using (4.3) from lemma 4.1.

6.3.3 AR-NARX Model: Incorporating a noise model

Consider the AR-NARX model (6.2), with $g(\cdot)$ parameterized as in (6.5). With the inclusion of an AR(1) noise correlation model, the following regularized optimization problem is formulated:

$$\min_{\boldsymbol{w},b,r_i,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 \tag{6.10}$$

$$\text{s.t.} \begin{cases} y_i = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_i) + b + e_i, & i = 2, \ldots, N, \\ e_i = \rho e_{i-1} + r_i, & i = 2, \ldots, N, \end{cases}$$

where γ is a regularization constant and the noise model coefficient ρ is a tuning parameter satisfying $|\rho| < 1$ (invertibility condition of the process). The solution is obtained using (5.8) from lemma 5.1.

6.3.4 PL-AR-NARX Model: Combining it all

Consider now the PL-AR-NARX model, with $g(\cdot)$ parameterized as in (6.7). With the inclusion of an AR(1) noise correlation model, the following

regularized optimization problem is formulated:

$$\min_{\boldsymbol{w},b,r_i,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 \tag{6.11}$$
$$\text{s.t.} \quad \begin{cases} y_i = \boldsymbol{\beta}^T\boldsymbol{z}_{A,i} + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{z}_{B,i}) + b + e_i, \quad i = 2,\ldots,N, \\ e_i = \rho e_{i-1} + r_i, \quad i = 2,\ldots,N, \end{cases}$$

where γ is a regularization constant and the noise model coefficient ρ is a given tuning parameter satisfying $|\rho| < 1$. By eliminating e_i, the following problem is formulated:

$$\min_{\boldsymbol{w},b,r_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 \tag{6.12}$$
$$\text{s.t.} \quad y_i = \rho y_{i-1} + \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{z}_{B,i}) - \rho\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{z}_{B,i-1})$$
$$+b(1-\rho) + \boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho\boldsymbol{z}_{A,i-1}) + r_i, \quad i = 2,\ldots,N.$$

The solution is formalized in the following lemma.

Lemma 6.1. *Given a positive definite kernel function $K : \mathbb{R}^{N_b} \times \mathbb{R}^{N_b} \to \mathbb{R}$, the solution to (6.12) is given by the dual problem*

$$\left[\begin{array}{c|c|c} \mathbf{0}_{N_a\times N_a} & \mathbf{0}_{N_a\times 1} & \tilde{\boldsymbol{Z}}^T \\ \hline \mathbf{0}_{1\times N_a} & 0 & \mathbf{1}^T \\ \hline \tilde{\boldsymbol{Z}} & \mathbf{1} & \boldsymbol{\Omega}^{(\rho)} + \gamma^{-1}\mathbf{I} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\beta} \\ \hline b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_{N_a\times 1} \\ \hline 0 \\ \hline \tilde{\boldsymbol{y}} \end{array}\right], \tag{6.13}$$

where $\tilde{\boldsymbol{Z}} = [\boldsymbol{z}_{A,2}^T - \rho\boldsymbol{z}_{A,1}^T; \cdots; \boldsymbol{z}_{A,N}^T - \rho\boldsymbol{z}_{A,N-1}] \in \mathbb{R}^{(N-1)\times N_a}$ is the matrix of linear regressors; $\tilde{\boldsymbol{y}} = [y_2 - \rho y_1, \ldots, y_N - \rho y_{N-1}]^T$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{N-1}]^T$, and $\boldsymbol{\Omega}^{(\rho)}$ is the kernel matrix with entries $\boldsymbol{\Omega}_{ij}^{(\rho)} = K(\boldsymbol{z}_{B,i+1}, \boldsymbol{z}_{B,j+1}) - \rho K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,j+1}) - \rho K(\boldsymbol{z}_{B,i+1}, \boldsymbol{z}_{B,j}) + \rho^2 K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,j})$, $\forall i,j = 1\ldots,N-1$.

Proof: Consider the Lagrangian of problem (6.12)

$$\mathscr{L}(\boldsymbol{w}, b, r_i, \boldsymbol{\beta}; \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 - \sum_{i=2}^{N}\alpha_{i-1}[\boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho\boldsymbol{z}_{A,i-1})$$
$$+ \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{z}_{B,i}) - \rho\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{z}_{B,i-1}) + \rho y_{i-1} - y_i - r_i],$$

where $\alpha_j \in \mathbb{R}, j = 1,\ldots,N-1$ are the Lagrange multipliers. Taking the optimality conditions $\frac{\partial\mathscr{L}}{\partial\boldsymbol{w}} = 0$, $\frac{\partial\mathscr{L}}{\partial b} = 0$, $\frac{\partial\mathscr{L}}{\partial r_i} = 0$, $\frac{\partial\mathscr{L}}{\partial\alpha_j} = 0$, $\frac{\partial\mathscr{L}}{\partial\boldsymbol{\beta}} = 0$ yields

$$\boldsymbol{w} \;=\; \sum_{k=2}^{N}\alpha_{(k-1)}[\boldsymbol{\varphi}(\boldsymbol{z}_{B,k}) - \rho\boldsymbol{\varphi}(\boldsymbol{z}_{B,k-1})],$$

$$
\begin{aligned}
r_i &= \alpha_{i-1}/\gamma, \quad i = 2, \ldots, N, \\
0 &= \sum_{k=1}^{N-1} \alpha_k, \\
0 &= \sum_{i=2}^{N} \alpha_{i-1}(\boldsymbol{z}_{A,i} - \rho \boldsymbol{z}_{A,i-1}), \\
y_i &= \rho y_{i-1} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,i}) - \rho \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,i-1}) \\
&\quad + b(1-\rho) + \boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho \boldsymbol{z}_{A,i-1}) + r_i, \quad i = 2, \ldots, N. \quad (6.14)
\end{aligned}
$$

With application of Mercer's theorem [129] $\boldsymbol{\varphi}(\boldsymbol{z}_{B,i})^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,j}) = K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,j})$ with a positive definite kernel K, $\boldsymbol{w}$ and r_i can be eliminated, yielding

$$
\begin{aligned}
y_i - \rho y_{i-1} = \sum_{k=2}^{N} \alpha_{k-1} [K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,k}) - \rho K(\boldsymbol{x}_{B,i-1}, \boldsymbol{x}_{B,k}) - \rho K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,k-1}) \\
+ \rho^2 K(\boldsymbol{z}_{B,i-1}, \boldsymbol{z}_{B,k-1})] + b + \boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho \boldsymbol{z}_{A,i-1}) + \frac{\alpha_{k-1}}{\gamma}, \\
i = 1, \ldots, N-1. \quad (6.15)
\end{aligned}
$$

Building the kernel matrix $\boldsymbol{\Omega}_{ij}^{(\rho)}$ and writing the equations in matrix notation gives the final system (6.13). ■

The estimated model in dual space becomes

$$
\hat{y}_t = \rho y_{t-1} + h(\boldsymbol{z}_{B,t}) - \rho h(\boldsymbol{z}_{B,t-1}) + \boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho \boldsymbol{z}_{A,i-1}), \quad (6.16)
$$

where $h(\boldsymbol{z}_{B,t})$ is

$$
h(\boldsymbol{z}_t) = \sum_{i=2}^{N} \alpha_{i-1} \left[K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,t}) - \rho K(\boldsymbol{z}_{B,i-1}, \boldsymbol{z}_{B,t}) \right] + b. \quad (6.17)
$$

Remark 6.1. [Including Symmetry] *For all model structures above it is straightforward to impose symmetry to the nonlinear function $g(\cdot)$ in (6.7) using the equivalent kernel function (3.9) in each case. In this way, symmetry is already embedded at the kernel level. This makes the entire methodology for (AR)-NARX model estimation much more modular.*

A summary of the different nonlinear model structures and representations is given in Table 1.

NARX Model	
Primal	$\hat{y}_t = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_t) + b$
Dual	$\hat{y}_t = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{z}_i, \boldsymbol{z}_t) + b$
AR-NARX Model	
Primal	$\hat{y}_t = \rho y_{t-1} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_t) - \rho \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{t-1}) + (1-\rho) b$
Dual	$\hat{y}_t = \rho y_{t-1} + h(\boldsymbol{z}_t) - \rho h(\boldsymbol{z}_{t-1})$ with $h(\boldsymbol{z}_t) = \sum_{i=2}^{N} \alpha_{i-1} \left[K(\boldsymbol{z}_i, \boldsymbol{z}_t) - \rho K(\boldsymbol{z}_{i-1}, \boldsymbol{z}_t) \right] + b$
PL-NARX Model	
Primal	$\hat{y}_t = \boldsymbol{\beta}^T \boldsymbol{z}_{A,t} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,t}) + b$
Dual	$\hat{y}_t = \boldsymbol{\beta}^T \boldsymbol{z}_{A,t} + \sum_{i=1}^{N} \alpha_i K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,t})$
PL-AR-NARX Model	
Primal	$\hat{y}_t = \rho y_{t-1} + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,t}) - \rho \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_{B,t-1}) + b(1-\rho)$ $+ \boldsymbol{\beta}^T (\boldsymbol{z}_{A,t} - \rho \boldsymbol{z}_{A,t-1})$
Dual	$\hat{y}_t = \rho y_{t-1} + h(\boldsymbol{z}_{B,t}) - \rho h(\boldsymbol{z}_{B,t-1}) + \boldsymbol{\beta}^T (\boldsymbol{z}_{A,i} - \rho \boldsymbol{z}_{A,i-1})$ with $h(\boldsymbol{z}_{B,t}) = \sum_{i=2}^{N} \alpha_{i-1} \left[K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,t}) - \rho K(\boldsymbol{z}_{B,i-1}, \boldsymbol{z}_{B,t}) \right] +$ b

Table 6.1: *Summary of Nonlinear Model Structures and Representations using LS-SVMs.*

6.3.5 Links with other model representations

In general, a Hammerstein single-input-single-output (SISO) model

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \sum_{j=1}^{q} b_j h(u_t) + e_t,$$

contains a static nonlinearity h applied over the input u_t. The Generalized Hammerstein model extends the concept to include a nonlinear finite impulse response (NFIR) formulation instead of a static nonlinearity,

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \sum_{j=1}^{q} b_j h(u_t, u_{t-1}, \ldots, u_{t-k}) + e_t.$$

In these formulations it is possible to apply a PL-NARX structure to identify the coefficients of the linear part and the nonlinear total component by an obvious definition of f in (4.1) as

$$f(u_t) = \sum_{j=1}^{q} b_j h(u_t),$$

in the first case, and

$$f(u_t, \ldots, u_{t-k}) = \sum_{j=1}^{q} b_j h(u_t, u_{t-1}, \ldots, u_{t-k}),$$

in the second case. However, with the exception of simple cases ($q = 1$), the identification of f does not translate directly to an identification of h; for a detailed identification of the function h eventually an ad-hoc structure is required [45] where further restrictions are imposed to the function f.

Consider the AR-NARX model described in (6.2). Interesting links with existing and well known model representations can be established for the case where $\boldsymbol{z}_i$ does not contain past values of the output, that is, the nonlinear function $g(\boldsymbol{z}_i)$ is a static nonlinearity. Considering $\boldsymbol{z}_i$ as an exogenous input, the model structure

$$A(z^{-1})y_t = A(z^{-1})g(\boldsymbol{z}_t) + r_t, \tag{6.18}$$

is equivalent to a Hammerstein system [18]

$$y_t = \sum_{i=1}^{r} c_i y_{t-1} + \sum_{i=0}^{s} d_i g(\boldsymbol{z}_{t-i}) + r_t, \tag{6.19}$$

where the order is given by the order of the AR(q) residual process ($r = s = q$), and the following conditions on the coefficients hold: $c_i = -a_i$, $d_i = a_i$, $i = 1, \ldots, q$ and $d_0 = 1$.

Alternatively, additional insights into the model structure can be obtained when considering the model formulation as a state-space description. Consider the case for $q = 1$ of model (6.2), described as

$$\begin{cases} e_{t+1} &= \rho e_t + r_{t+1} \\ y_t &= e_t + g(\boldsymbol{z}_t). \end{cases} \tag{6.20}$$

The AR(1) process representation corresponds to the state equation. In this interpretation, e_t corresponds to the unobserved state of the system, r_{t+1} is the process noise, and ρ is the parameter for the state equation of this system. The output equation consists of the state e_k with coefficient equal to 1, and an input described as a nonlinear function of the vector $\boldsymbol{z}_t$. The above description gives explicit expressions for optimal prediction, where not only the nonlinear function g has to be approximated, but also the corresponding state should be predicted as well. With this interpretation, the optimal predictor for $t + 1$ given the information up to time t can be easily obtained in terms of both the predictors of the future state $t_{t+1|t}$ and the output $y_{t+1|t}$ by means of, for example, Kalman filter applied to (6.20), and is equivalent to the predictor obtained from the system (5.19) for the case of a static nonlinearity.

6.4 Model estimation in Primal Space

All models described above are expressed in terms of the dual solution, requiring solving a linear system of size $(N + N_a) \times (N + N_a)$ (for a NARX model, $N_a = 0$). This system is obtained with the application of Mercer's theorem, without having to compute the nonlinear mapping $\boldsymbol{\varphi}$ explicitly. However, for large sample sizes this may become too time consuming or simply unpractical. In such a case, the models can be estimated in primal space using the Nyström approximation (2.22) and (2.23). The approximation has to be computed with the generic kernel matrix $\boldsymbol{\Omega}^{\text{eq}}$ with entries $\boldsymbol{\Omega}^{\text{eq}}_{ij} = K^{\text{eq}}_{\text{model}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where the kernel function $K^{\text{eq}}_{\text{model}}$ can be defined for each of the model structures, as listed in Table 6.2, and where the vector $\boldsymbol{x}$ represents $\boldsymbol{z}$ or $\boldsymbol{z}_B$ depending on the model.

From a kernel matrix $\boldsymbol{\Omega}^{\text{eq}}$ evaluated over a data subsample of fixed-size M,

$K^{\text{eq}}_{\text{narx}}(\boldsymbol{x}_i, \boldsymbol{x}_j) =$	$K(\boldsymbol{z}_i, \boldsymbol{z}_j)$
$K^{\text{eq}}_{\text{ar-narx}}(\boldsymbol{x}_i, \boldsymbol{x}_j) =$	$K(\boldsymbol{z}_{i+1}, \boldsymbol{z}_{j+1}) - \rho K(\boldsymbol{z}_i, \boldsymbol{z}_{j+1}) - \rho K(\boldsymbol{z}_{i+1}, \boldsymbol{z}_j) + \rho^2 K(\boldsymbol{z}_i, \boldsymbol{z}_j)$
$K^{\text{eq}}_{\text{pl-narx}}(\boldsymbol{x}_i, \boldsymbol{x}_j) =$	$K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,j})$
$K^{\text{eq}}_{\text{pl-ar-narx}}(\boldsymbol{x}_i, \boldsymbol{x}_j) =$	$K(\boldsymbol{z}_{B,i+1}, \boldsymbol{z}_{B,j+1}) - \rho K(B, \boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,j+1}) - \rho K(\boldsymbol{z}_{B,i+1}, \boldsymbol{z}_{B,j}) + \rho^2 K(\boldsymbol{z}_{B,i}, \boldsymbol{z}_{B,j})$

Table 6.2: *Equivalent kernel function* $K^{\text{eq}}_{\text{model}}$ *for the different model structures. The vector* $\boldsymbol{x}$ *represents* $\boldsymbol{z}$ *or* $\boldsymbol{z}_B$*, depending on the model structure. The finite dimensional approximation for the feature map can be computed using the Nyström method with any of these kernel functions. This provides a modular approach for large scale nonlinear regression problems.*

the approximation $\hat{\boldsymbol{\varphi}}$ is obtained using (2.22) for each of the components. Let $\hat{\boldsymbol{\varphi}}^{\text{model}}(\boldsymbol{z}_t)$ be the approximation of the feature map for a datapoint $\boldsymbol{z}_t$ for a given model structure. The model can now be estimated in primal space directly using ridge regression techniques with regularization parameter γ. In other words, the problem can be solved by minimizing a regularized least-squared cost function as follows.

- For the NARX model in Primal Space, the solutions $\boldsymbol{w}, b$ are obtained from

$$\min_{\boldsymbol{w},b,e_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \tag{6.21}$$

$$\text{s.t.} \quad y_i = \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{narx}}(\boldsymbol{z}_i) + b + e_i, \quad i = 1, \ldots, N.$$

  With the explicit expression for $\hat{\boldsymbol{\varphi}}^{\text{narx}}(\boldsymbol{z}_i)$, the model is solved in primal

space by eliminating e_i from (6.21),

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{narx}}(\boldsymbol{z}_i) - b)^2. \tag{6.22}$$

- For the AR-NARX model, the solutions $\boldsymbol{w}, b$ are obtained from

$$\min_{\boldsymbol{w},b,r_i} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 \tag{6.23}$$

$$\text{s.t.} \quad y_i = \rho y_{i-1} - \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{ar-narx}}(\boldsymbol{z}_i) + b + r_i, \quad i = 2,\ldots,N.$$

where the autocorrelation structure for the nonlinear function is embedded into the evaluation of $\hat{\boldsymbol{\varphi}}^{\text{ar-narx}}(\boldsymbol{z}_i)$ computed from the kernel matrix $\boldsymbol{\Omega}^{\text{eq}}$. With the explicit expression for $\hat{\boldsymbol{\varphi}}^{\text{ar-narx}}(\boldsymbol{z}_i)$, the model is solved in primal space by eliminating r_i from (6.23),

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N}(y_i - \rho y_{i-1} - \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{ar-narx}}(\boldsymbol{z}_i) - b)^2. \tag{6.24}$$

- For the PL-NARX model, the solutions $\boldsymbol{\beta}, \boldsymbol{w}, b$ are obtained from

$$\min_{\boldsymbol{w},b,e_i,\boldsymbol{\beta}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \tag{6.25}$$

$$\text{s.t.} \quad y_i = \boldsymbol{\beta}^T\boldsymbol{z}_{A,i} + \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{pl-narx}}(\boldsymbol{z}_{B,i}) + b + e_i, \quad i = 1,\ldots,N,$$

With the explicit expression for $\hat{\boldsymbol{\varphi}}^{\text{pl-narx}}(\boldsymbol{z}_{B,i})$, the model is solved in primal space by eliminating e_i from (6.25),

$$\min_{\boldsymbol{w},b,\boldsymbol{\beta}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N}(y_i - \boldsymbol{\beta}^T\boldsymbol{z}_{A,i} - \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{pl-narx}}(\boldsymbol{z}_{B,i}) - b)^2. \tag{6.26}$$

- Finally, for the PL-AR-NARX model structure, the solutions $\boldsymbol{\beta}, \boldsymbol{w}, b$ are obtained from

$$\min_{\boldsymbol{w},b,r_i,\boldsymbol{\beta}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N} r_i^2 \tag{6.27}$$

$$\text{s.t.} \quad y_i = \rho y_{i-1} + \boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho\boldsymbol{z}_{A,i-1}) + \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\text{pl-ar-narx}}(\boldsymbol{z}_{B,i}) + b + r_i,$$

for $i = 2, \ldots, N$. With the explicit expression for $\hat{\boldsymbol{\varphi}}^{\mathrm{pl-ar-narx}}(\boldsymbol{z}_{B,i})$, the model is solved in primal space by eliminating r_i from (6.27),

$$\min_{\boldsymbol{w},b,\boldsymbol{\beta}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=2}^{N}(y_i - \rho y_{i-1} - \boldsymbol{\beta}^T(\boldsymbol{z}_{A,i} - \rho\boldsymbol{z}_{A,i-1}) - \boldsymbol{w}^T\hat{\boldsymbol{\varphi}}^{\mathrm{pl-ar-narx}}(\boldsymbol{z}_{B,i}) - b)^2. \quad (6.28)$$

6.5 Examples

This section shows some illustrative examples for the estimation of NARX models using LS-SVM. Most of the examples are implemented in primal space, that is, the initial sample of size M is selected using the quadratic Renyi entropy criterion. On each example, an RBF kernel is used and the parameters σ and γ are found by 10-fold cross validation over the corresponding training sample.

6.5.1 Examples for NARX Models

1. *Time Series forecasting.* The laser example of the Santa Fe competition [134] of time series prediction is used. Given 1000 historical datapoints, the goal is to predict the next 100 values using an iterative simulation procedure. This is, predict the first point out of sample, then use this prediction to compute the next prediction, and so on. A NARX model is estimated of the form $\hat{y}_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-p})$, selecting $p = 50$. In this setting, the training sample size is $N = 900$. The subsampling technique is implemented with a sample size of $M = 200$ datapoints.

2. *Input-Output Model.* This example is taken from the DaISy [23] datasets. The process is a liquid-saturated steam heat exchanger, where water is heated by pressurized saturated steam through a copper tube. The output variable y_t is the outlet temperature, and the input variable u_t is the flow rate. A NARX model of the form $\hat{y}_t = f(y_{t-1}, \ldots, y_{t-p}, u_{t-1}, \ldots, u_{t-p})$ with $p = 5$ is estimated, with a sample size of $N = 1800$, and $M = 200$ initial support vectors. Out-of-sample predictions are computed for the next 200 values. This is an example of a larger dataset, where working with the full sample N

would add computational cost. For comparison, the result of the same model under a linear estimation with the same p is reported.

Each one of these applications is independently trained and estimated for the following cases:

- *Case I.* Using the full sample of size N to obtain the optimal hyperparameter, define the regressors and the final estimation.
- *Case II.* Using only a fixed-size subsample for finding the hyperparameter, the regressors and the final model.

The goal is to perform the training and estimation procedures independently, using only the available information for each Case. In other words, no information from Case I is used in Case II, as for large scale problems the only feasible way to proceed is to use the fixed-size method. The results reported on each case are:

1. The optimal σ found by minimizing the cross-validation MSE;
2. The value of M, the number of support vectors selected for the regression in primal space;
3. The MSE (mean squared error) both in-sample and out-of-sample.

The results are summarized in Table 6.3 and accompanying figures. In general, we observe satisfactory results on the performance of the models. It is important to notice that the good performance of the cases when $M \ll N$ is due not only to the quality of the Nyström approximation, but also to the appropriate selection of the support vectors by means of the Renyi quadratic entropy maximization. In the Laser problem, the way the support vectors spread around the zones where important changes on the levels of the series are taking place is remarkable. With this selection of the support vectors, the results obtained for the iterative prediction are very close to those obtained using the entire sample, as seen in Figures 6.1 and 6.2 respectively. Figure 6.3 shows the evolution of the entropy during the support vector selection for the Laser and the Heat-Exchanger examples. The performance of the methodology on the 200 predicted values for the Heat-Exchanger example is shown in Figure 6.4.

Problem	σ	M	MSE_{IN}	MSE_{OUT}
Laser (Santa Fe)				
Case I	5.3	900	0.01	0.05
Case II	4.2	200	0.02	0.06
Heat Exchanger				
Case I	5.8	1800	0.04	0.06
Case II	4.7	200	0.04	0.07
Linear (same p)	-	1800	0.04	0.23

Table 6.3: *Performance of the estimations in primal space. Case I uses the full sample ($M = N$), and Case II uses a fixed-size ($M \ll N$) version.*

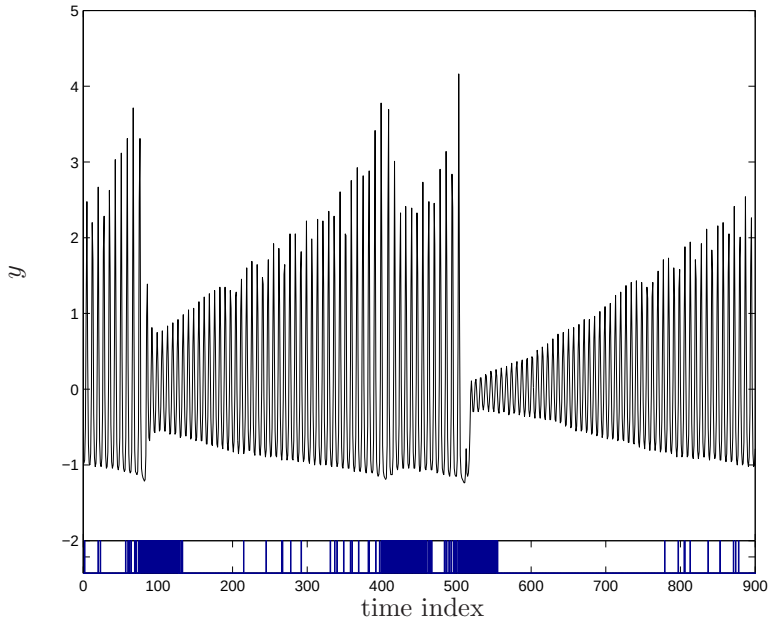

Figure 6.1: *Training sample for the Santa Fe Laser data. Case I estimations uses of the full sample. The 200 selected support vectors can be visualized in terms of their time index position, indicated by the dark bars at the bottom. Remarkably, the selected support vectors are placed around critical transition regions of the dataset.*

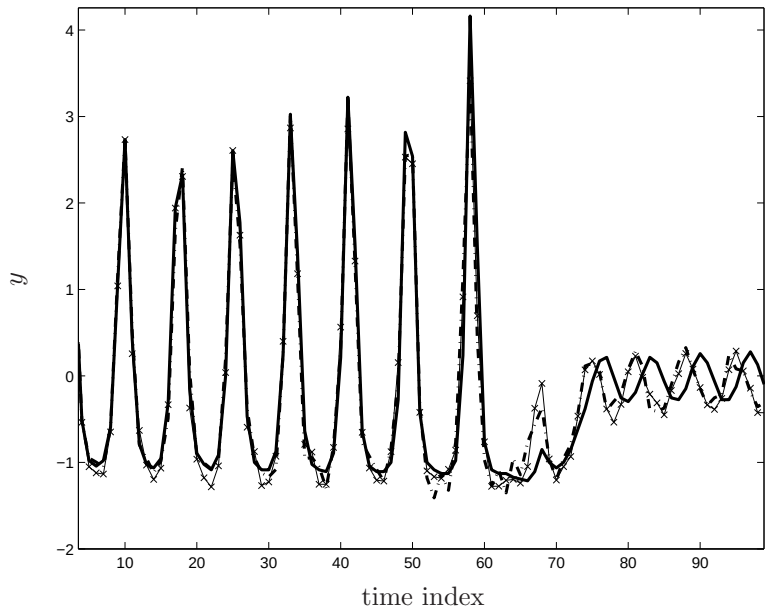

Figure 6.2: *Iterative prediction for the Laser example, for Case I ('-x' line), Case II ('-.' line), and 'true' values (full line).*

6.5.2 Examples for Partially Linear Structures

The test cases for partially linear parameterizations are defined as follows:

- **Example PL-I: Autoregression with linear and nonlinear components.** The model to be estimated is of the form $y_t = a_1 y_{t-1} + a_2 y_{t-2} + \mathrm{sinc}(y_{t-3}) + e_t$, where the true values are $a_1 = 0.6, a_2 = 0.3$; e_t is a Gaussian white noise of variance 0.02.
- **Example PL-II: Hammerstein Model.** The true model is $y_t = a_1 y_{t-1} + a_2 y_{t-2} + a_3 y_{t-3} + b_1 \mathrm{sinc}(u_{t-1}) + b_2 \mathrm{sinc}(u_{t-2}) + e_t$, with $a_1 = 0.6, a_2 = 0.2, a_3 = 0.1, b_1 = 0.4, b_2 = 0.2$. The input u_t comes from a Gaussian distribution with mean 0 and variance 2, and e_t is a Gaussian noise with variance 0.1.
- **Example PL-III: Generalized Hammerstein Model.** The true model is a Generalized Hammerstein model $y_t = a_1 y_{t-1} + a_2 y_{t-2} + \arctan(u_t) u_{t-1}^2 + \varepsilon_t$, with $a_1 = -0.6, a_2 = -0.1$ and the input series

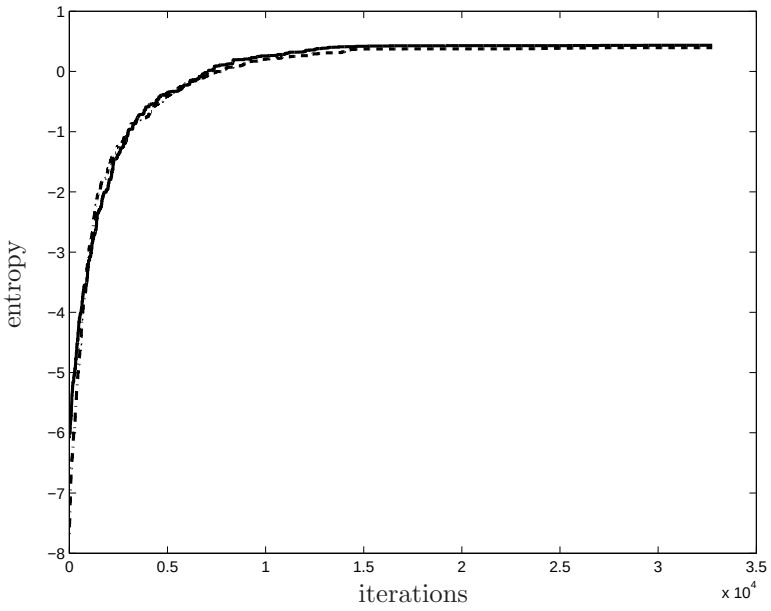

Figure 6.3: *Convergence of the Renyi entropy for the support vector selection in the Laser problem (full line) and the Heat Exchanger problem ('-.' line). Values have been normalized for comparison.*

is generated by $u_t = b_1 u_{t-1} + \varepsilon_{t-1} + \varepsilon_{t-2}$ where ε_t is Gaussian noise with variance 1 (this example is taken from [27]).

It worth noting that although the regressors contained in the linear part might be correlated with the regressors under the nonlinear part, they are neither identical nor perfectly related to each other. Therefore, there linear part is uniquely represented. The out-of-sample performance, on an iterative basis (simulation mode) is examined for the models. 1,000 datapoints are generated and the first 400 are dismissed to remove any transient effect. 500 datapoints are then used for training, and the performance is measured over the next 50 out-of-sample points running the model iteratively in simulation mode, each time using past predictions as inputs to produce the next forecasts.

The models are estimated where the hyperparameters are selected using a 10-fold crossvalidation. The results are reported in Table 6.4 in terms of the estimation results and out-of-sample performance. The MSE obtained

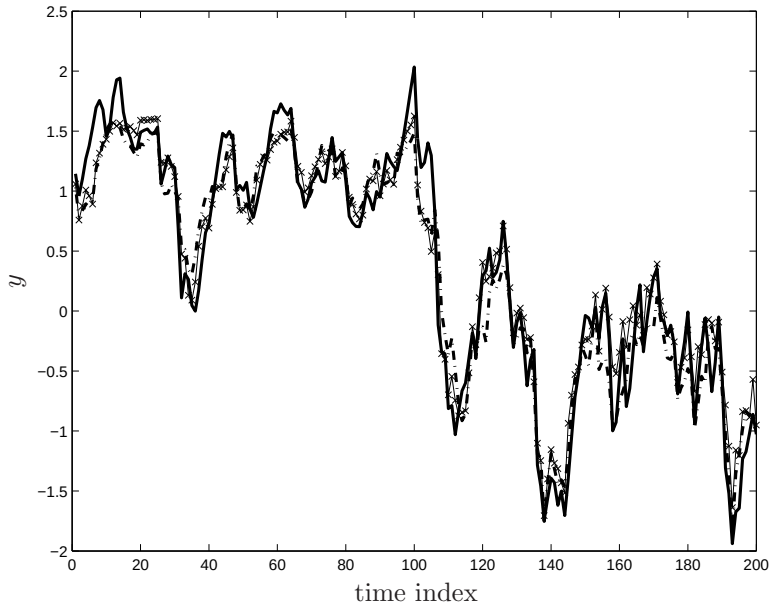

Figure 6.4: *Iterative prediction for the Heat-Exchanger example, for Case I ('-x' line), Case II ('-.' line), and 'true' values (full line).*

in the out-of-sample exercise (MSE simulation) is very close to the MSE level obtained within the training procedure by 10-fold cross validation (CV-MSE). At the same time, the linear parameters for each model are identified successfully. The out-of-sample iterative prediction is computed by sequentially using past predictions as new inputs for the autoregressive part, in simulation mode [76]. All models perform substantially well, as shown in Figure 6.5 for Example PL-I (top), Example PL-II (middle) and Example PL-III (bottom), for the comparison between the predictions and the true values for the next 50 points out-of-sample.

6.5.3 Examples for Models with Symmetry

In this subsection, examples of imposing symmetry to the LS-SVM are presented for two cases of chaotic time series. In each example, an RBF kernel is used and the parameters σ and γ are found by 10-fold cross validation over the corresponding training sample. The results using the standard LS-SVM are compared to those obtained with the symmetry-

	Estimates			MSE	
	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	CV (train)	Simulation
Example PL-I	0.598	0.302	-	0.006	0.005
Example PL-II	0.597	0.195	0.11	0.007	0.010
Example PL-III	-0.592	-0.098	-	1.19	1.18

Table 6.4: *Parameter estimates, MSE (CV-training and Simulation) for Example PL-I (NAR), Example PL-II (Hammerstein) and Example PL-III (Generalized Hammerstein).*

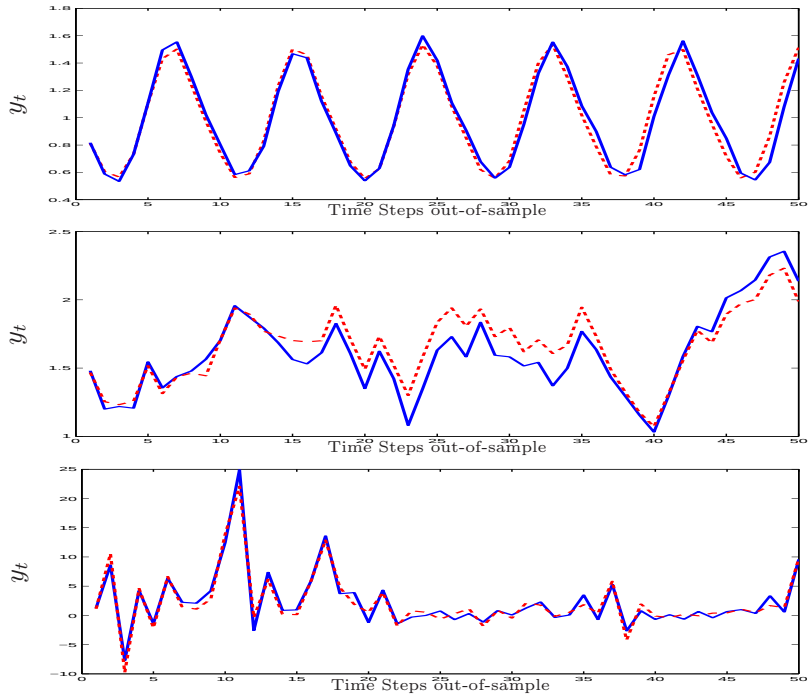

Figure 6.5: *Simulated (dashed) and Observed (solid) values for the next 50 time steps out-of-sample for Example PL-I (top), Example PL-II (middle) and Example PL-III (bottom).*

constrained LS-SVM (S-LS-SVM) from (3.2). The examples are defined in such a way that there are not enough training datapoints in every region of the relevant space. Therefore, it is very difficult for a black-box model to "learn" about the symmetry just by using the available information. The examples are compared in terms of the performance in the training sample (cross-validation mean squared error, MSE-CV) and the generalization performance (MSE out of sample, MSE-OUT). In each case, a NAR(X) black-box model is formulated:

$$y_t = g(y_{t-1}, y_{t-2}, \ldots, y_{t-p}) + e_t$$

where g is to be identified by LS-SVM and S-LS-SVM. The order p is selected during the cross-validation process as an extra parameter. After each model is estimated, it is used in simulation mode, where the future predictions are computed with the estimated model $\hat{\varphi}$ using past predictions:

$$\hat{y}_t = \hat{g}(\hat{y}_{t-1}, \hat{y}_{t-2}, \ldots, \hat{y}_{t-p}).$$

1. *Lorenz attractor.* This example is taken from [1]. The $x-$coordinate of the Lorenz attractor is used as an example of a time series generated by a dynamical system. A sample of 1000 datapoints is used for training, corresponding to an unbalanced sample over the evolution of the system, shown on Figure 6.6 as a time-delay embedding. Figure 6.7 (top) shows the training sequence (thick line) and the future evolution of the series (test zone). Figure 6.7 (bottom) shows the simulations obtained from both models on the test zone. Results are presented in Table 6.5. Clearly the S-LS-SVM can simulate the system for the next 500 timesteps, far beyond the 100 points that can be simulated by the LS-SVM.

2. *Multi-scroll attractors.* This dataset was used for the 1998 K.U.Leuven Time Series Prediction Competition [117]. This series is generated by

$$\dot{\boldsymbol{x}} = h(\boldsymbol{x}) \tag{6.29}$$

	LS-SVM	S-LS-SVM
MSE-CV	3.41×10^{-4}	1.62×10^{-4}
MSE-OUT	52.057	0.085

Table 6.5: *Performance of LS-SVM and S-LS-SVM on the Lorenz attractor data.*

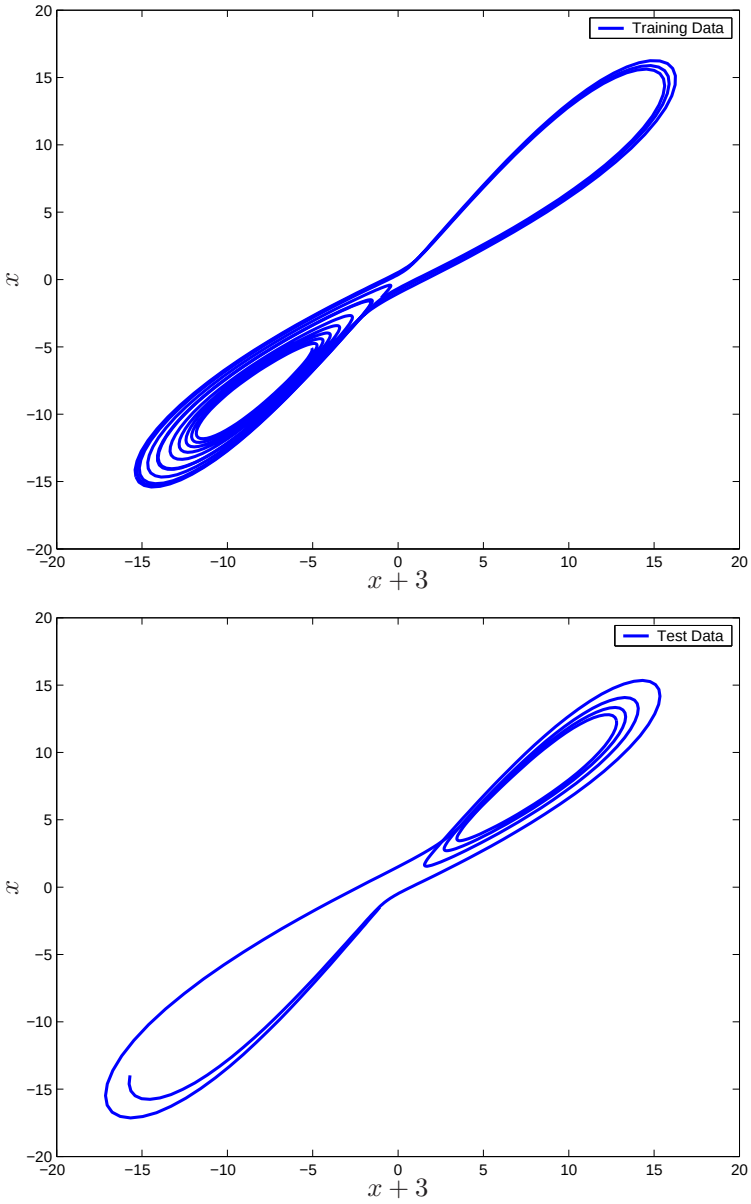

Figure 6.6: *Training (top) and test (bottom) series from the $x-$coordinate of the Lorenz attractor.*

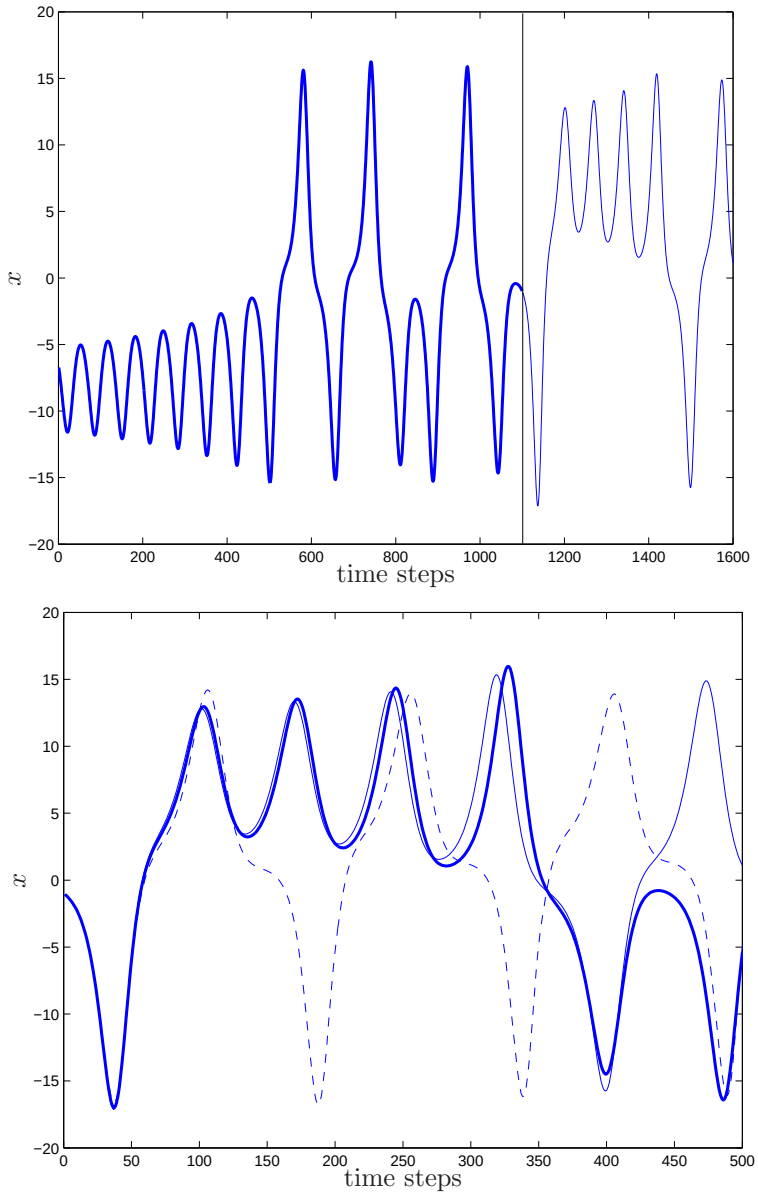

Figure 6.7: *(Top) The series from the $x-$coordinate of the Lorenz attractor, part of which is used for training (thick line). (Bottom) Simulations with LS-SVM (dashed line), S-LS-SVM (thick line) compared to the actual values (thin line).*

$$\boldsymbol{y} \quad = \quad \boldsymbol{W} \tanh\left(\boldsymbol{V}\boldsymbol{x}\right)$$

where h is the multi-scroll equation, $\boldsymbol{x}$ is the 3-dimensional coordinate vector, and $\boldsymbol{W}, \boldsymbol{V}$ are the interconnection matrices of the nonlinear function (a 3-units multilayer perceptron, MLP). This MLP function hides the underlying structure of the attractor [86]. A training set of 2,000 points is available for model estimation, shown on Figure 6.8, and the goal is to predict the next 200 points out of sample. The winner of the competition followed a complete methodology involving local modelling, a specialized parameters tuning procedure through many-steps-ahead cross-validation, and the exploitation of the symmetry properties of the series by flipping the series around the time axis.

Following the winner approach, both LS-SVM and S-LS-SVM are trained using 10-step-ahead cross-validation for hyperparameters selection. To illustrate the difference between both models, the out of sample MSE is computed considering only the first n simulation points, where $n = 20, 50, 100, 200$. It is important to emphasize that both models are trained using exactly the same methodology for order and hyperparameter selection; the only difference is the symmetry constraint for the S-LS-SVM case. Results are reported in Table 6.6. The simulations from both models are shown on Figure 6.9.

6.6 Conclusions

In the context of applied nonlinear system identification, it is possible to use the LS-SVM as an estimation method. This chapter shows that it is possible to build a modeling methodology around the central LS-SVM formulation,

	LS-SVM	S-LS-SVM
MSE-CV	0.15	0.11
MSE-OUT (1-20)	0.03	0.03
MSE-OUT (1-50)	0.05	0.03
MSE-OUT (1-100)	0.05	0.03
MSE-OUT (1-200)	0.64	0.24

Table 6.6: *Performance of LS-SVM and S-LS-SVM for the K.U.Leuven data.*

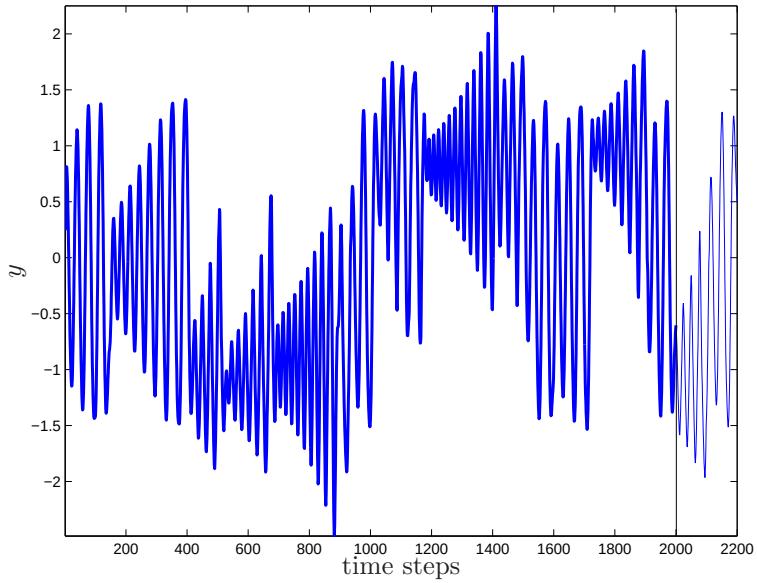

Figure 6.8: *Training sample (thick line) and future evolution (thin line) of the series from the K.U.Leuven Time Series Competition.*

taking the results of the previous chapters in a modular approach. This methodology has been developed to work with NARX and AR-NARX model structures, which can be parameterized as a fully nonlinear black-box model or in a partially linear form. For the case of the AR-NARX model structure, it has been shown to be equivalent to a very specific Hammerstein formulation. The derivations have been presented for both dual and primal formulations, with their corresponding representations and practical expressions for the equivalent kernel in each case. This leads to a powerful methodology for a modular modeling strategy, suitable for both large dimensional (dual space formulation) and large scale (primal space formulation) problems. Practical examples for chaotic time series show the effect of the selection of support vectors by means of the quadratic Renyi entropy, the satisfactory forecasting performance of the models, and the additional benefit of incorporating prior-knowledge of the problem at hand.

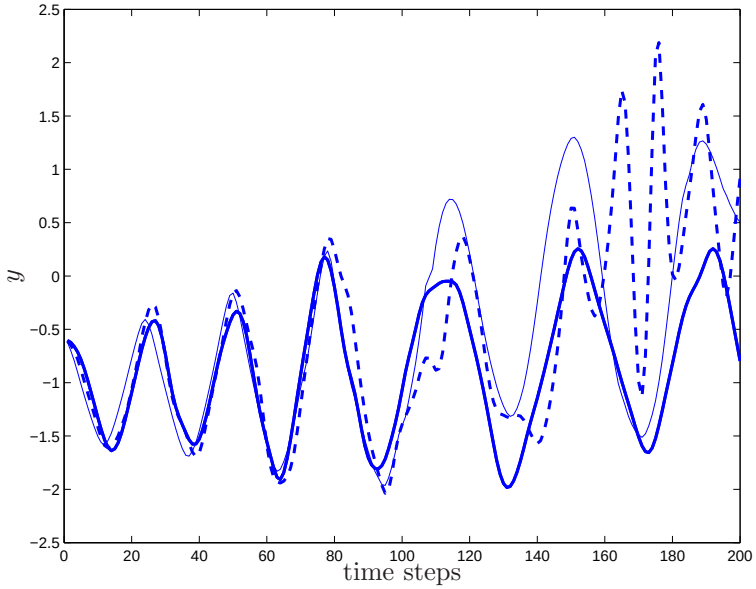

Figure 6.9: *Simulations with LS-SVM (dashed line), S-LS-SVM (thick line) compared to the actual values (thin line) for the next 200 points of the K.U.Leuven data.*

Chapter 7

Case Study: The SilverBox

This chapter presents an application to a large scale experimental case study. In the context of nonlinear system identification, we apply different variants of LS-SVM to the SilverBox dataset in the framework of a benchmark study. Starting from the dual representation of the LS-SVM, and using Nyström techniques, it is possible to compute an approximation of the nonlinear mapping to be used in the primal space. In this way, primal space based techniques as Ordinary Least Squares (OLS), Ridge Regression (RR) and Partial Least Squares (PLS) are applied to the same dataset together with the dual version of LS-SVM. The results obtained with black-box parameterizations are the best in this benchmark study. In addition, the results are further improved when using structured models with a partially linear formulation. This chapter is structured as follows. The description of the dataset and the modeling strategy is given on Section 7.1. The implementation using a nonlinear black-box model is described in Section 7.2. Further implementations, including symmetry and partially linear models are described in Section 7.3 and Section 7.4, respectively.

7.1 The SilverBox Benchmark Study

The SilverBox dataset gets its name from the physical device from where it originates. It is an electrical circuit that simulates a mechanical oscillatory system with damping [102]. The system is known to contain a cubic nonlinearity. Data is generated by using signals of different amplitude. The benchmark study was defined as follows. The data to be used for model estimation is generated using a constant amplitude. The data in which the models should be tested, on the contrary, is generated using an increasing amplitude. The important element of this design is that the final part of the test data contains a zone of larger amplitude than the data used for model estimation and selection. By putting together all data, the zones of different amplitudes gets the shape on an arrow. An initial plot of the output (the "arrow") is given in Figure 7.1. The results of the study formed the basis of a special session in the 14th IFAC NOLCOS (Nonlinear Control Systems) conference [30, 77, 92, 109, 131]. The data contains samples for input u_i and output y_i, with $i = 1, \ldots, N$, with $N = 131,072$ datapoints. The working strategy for using the data in terms of training, validation and testing is as follows:

- Training Sample: First half of the "body of the arrow", i.e. datapoints 40,001 to 85,000. Models are estimated using this part of the data. The mean squared error (MSE) of a one-step-ahead prediction can be computed directly using this training sample.

- Validation Sample: Second half of the "body of the arrow", datapoints 85,001 to the end. Having estimated the model parameters using the training sample, the model is validated using new datapoints. The MSE on the validation set is computed on a one-step-ahead basis. Model selection is based on the validation MSE.

- Test Sample: "Head of the arrow", datapoints 1 to 40,000. After defined the optimal model using the validation MSE, the prediction for the test set is generated. In this case, an iterative prediction is computed for the entire test set (each time using past predictions as inputs, using the estimated model in simulation mode). The MSE on the test set is computed.

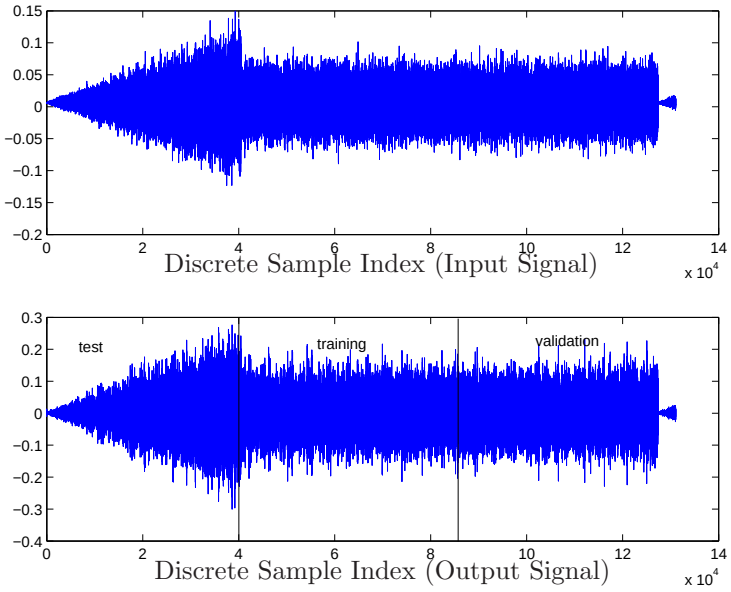

Figure 7.1: *Available data of the Silver Box identification problem. The zones for training, validation and testing are indicated.*

7.2 Nonlinear Black-Box approach

The general model structure is a NARX specification of the form $y_t = g(y_{t-1}, \ldots, y_{t-p}, u_t, u_{t-1}, \ldots, u_{t-p}) + e_t$. Exploratory analysis for estimating the order p is done based on the validation data. By using a black-box parameterization with LS-SVM as given by (6.5), the model is estimated using (6.8). No prior knowledge on the true system is available at this stage.

Given that there are approximately 40,000 datapoints to estimate this model, a Fixed-Size LS-SVM (2.26) formulation is used, leading to the estimation in primal space. For this analysis, the regression (6.8) is estimated in primal space using different traditional techniques. Ordinary Least Squares (OLS), Ridge Regression (RR) and Partial Least Squares (PLS) are applied. In the case of OLS, only $m < M$ components of $\hat{\boldsymbol{\varphi}}$ are used, and they are selected by looking at the eigenspectrum of the $M \times M$ kernel matrix $\boldsymbol{\Omega}_M$ used to build the approximation (2.22). In the case of RR [20], all components of $\hat{\boldsymbol{\varphi}}$ are used, and the regularization parameter γ

needs to be tuned accordingly. Finally, PLS involves an explicit construction of the set of regressors to be included in the model in order to take into account the information on the dependent variable and its correlation with the explanatory variables [99].

The different variants of LS-SVM to be applied are:

- LS-SVM in dual space (LS-SVM): for this method, a subsample of size 1000 is used for training, as using the full training sample is prohibitive.
- Fixed Size LS-SVM in primal space with OLS (FS-OLS), RR (FS-RR) and PLS (FS-PLS): for these methods, different numbers of support vectors are selected and the subsamples are selected by maximization of the quadratic entropy criterion.

The LS-SVM formulation requires to use a kernel matrix $\boldsymbol{\Omega}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Given that the nonlinear system is known to have a dominant linear behavior, we implemented not only the RBF kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2/\sigma^2)$, but also the polynomial kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T \boldsymbol{x}_j + c)^d$. Parameters σ, d, c and the regularization parameter γ are tuned based on the training-validation scheme.

7.2.1 Estimation and Model Selection

Using the definition of training and validation data described above, different lag orders and general parameters are tested. Each time the model is estimated using the training set and then evaluated in the validation set, always using the model to build predictions on a one-step-ahead basis. The combination of lag orders, kernel function and hyperparameters that gives the lowest MSE on the validation set (MSE_{val}) is selected.

An initial analysis using a linear ARX model with increasing lags of inputs and outputs, with the same training/validation scheme, shows that the MSE for the validation set can easily reach levels of 1.0×10^{-7}, corresponding to a root mean squared error (RMSE) of 3.2×10^{-4}. Figure 7.2 shows the MSE_{val} obtained when the number of lags varies from 5 to 40. This small error level at high lags can be a symptom of overfitting.

For the NARX models, Table 7.1 shows the best results (RMSE) achieved for each of the different techniques. It is important to remember that all

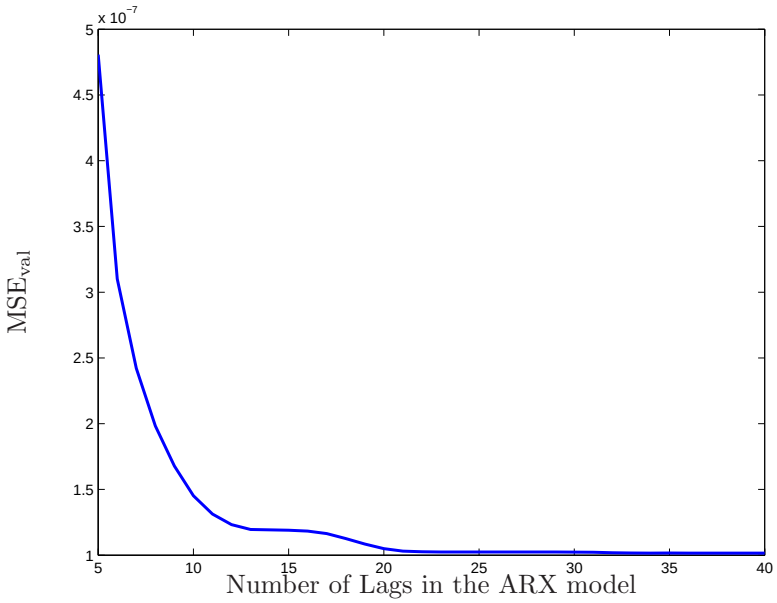

Figure 7.2: *Mean Squared Error in the validation set using a linear ARX model with increasing number of lags.*

techniques based on the Fixed-Size primal space version use the complete training/validation set; whereas the LS-SVM in dual space is limited to a subsample of 1000 points for training and validation. All RMSE figures are expressed in the original units of the data.

For all cases the polynomial kernel outperforms the RBF kernel, by up to 2 orders of magnitude. Although the RBF kernel is widely used, the dominant linear behavior of the data is better captured by the polynomial kernel. Additionally, the performance of the FS-RR and FS-PLS models with polynomial kernel is much better than the one obtained with the FS-OLS in the training/validation scheme.

The effect of selecting different numbers M of initial support vectors on the validation performance is shown in Table 7.2, for the FS-OLS version with polynomial kernel. clearly, the performance is improving marginally for $M > 500$. Therefore, taking into account practical considerations, $M = 500$ is chosen for the whole modeling exercise. The position of the selected 500 support vectors can be visualized in terms of the corresponding time index

Method	γ	Kernel	p	$\text{RMSE}_{\text{train}}$	RMSE_{val}
LS-SVM	10	Poly	5	5.1×10^{-5}	6.7×10^{-5}
	10	RBF	5	2.4×10^{-4}	2.7×10^{-4}
FS-OLS	-	Poly	7	3.6×10^{-4}	3.5×10^{-4}
	-	RBF	7	6.0×10^{-4}	1.3×10^{-3}
FS-RR	1000	Poly	10	2.3×10^{-4}	2.2×10^{-4}
	1000	RBF	10	6.0×10^{-4}	5.4×10^{-4}
FS-PLS	1000	Poly	10	2.31×10^{-4}	2.25×10^{-4}
	1000	RBF	10	1.1×10^{-3}	1.00×10^{-3}

Table 7.1: *Best models, based on the* RMSE_{val}*. For all cases shown,* $\sigma = 5.19p^{-1/2}$ *with p=number of lags (for RBF kernel);* $d = 3, c = 11$ *(for Polynomial kernel).*

position of the output data y_t. Figure 7.3 shows the output variable in the training set, and the dark bars at the bottom represent the position of the selected support vectors. The quite uniform distribution of the support vectors shows that this part of the dataset does not have critical transition regions or critical zones. Finally, the effect of the inclusion of different lags is tested for the NARX models, using lags from 2 to 10. Figure 7.4 shows the evolution of the MSE in the validation set for FS-RR (full line), FS-OLS (dash-dot) and FS-PLS (dashed).

7.2.2 Final Results on Test Data Set

After selecting the order of the models and the parameters involved, each one of the estimated models is used to build an iterative prediction (simulation mode, using only past predictions and input information) for the first 40,000 datapoints (the "head of the arrow"). As this is a completely unseen dataset, from the point of view of the modeling strategy, two types of error sources may be expected: the first one is due to the iterative (recurrent mode) nature of the simulations, so past errors can propagate to the next predictions. The second one is due to the fact that there are datapoints located beyond the amplitude range on which the models are trained, namely the wider zone of the "head of the arrow". The iterated prediction series is compared to the true values, and then this RMSE is computed on the test set ($\text{RMSE}_{\text{test}}$).

Table 7.3 shows the results obtained with the iterative prediction, for all

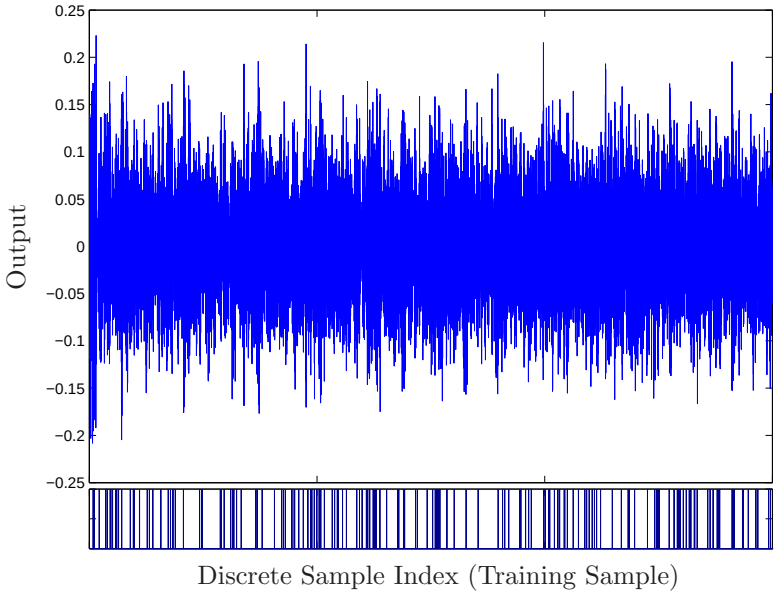

Figure 7.3: *(Top) Output training sample; (Bottom) The position, as time index, of the 500 selected support vectors is represented by dark bars*

Number of Support Vectors M	$\text{RMSE}_{\text{train}}$	RMSE_{val}
100	2.8×10^{-3}	2.5×10^{-3}
200	4.6×10^{-4}	4.4×10^{-4}
300	4.0×10^{-4}	3.8×10^{-4}
400	3.8×10^{-4}	3.7×10^{-4}
500	3.6×10^{-4}	3.5×10^{-4}
1000	3.5×10^{-4}	3.4×10^{-4}
1500	3.5×10^{-4}	3.4×10^{-4}

Table 7.2: *Effect of M on the performance of the FS-OLS estimator, measured by the root mean squared error (RMSE).*

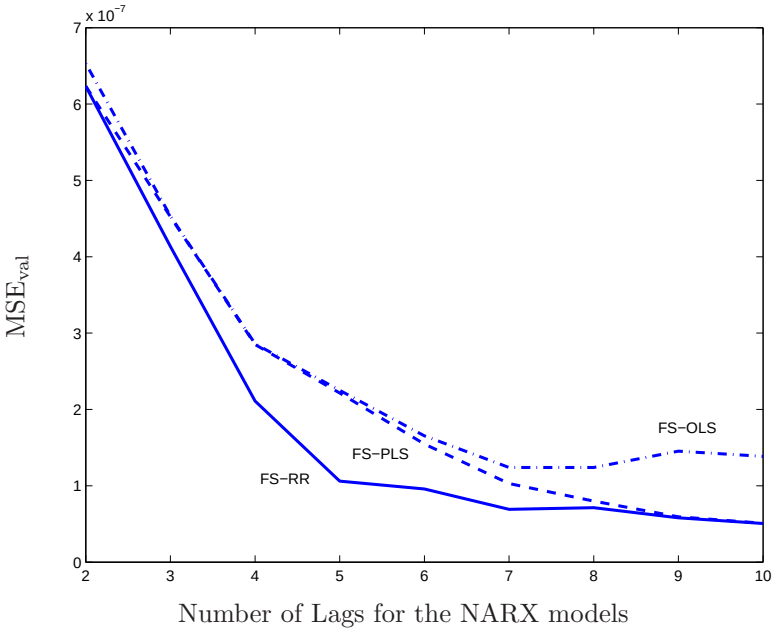

Figure 7.4: *MSE on the validation set obtained for FS-RR (full line), FS-PLS (dashed) and FS-OLS (dash-dot) using different number of lags.*

models, including the linear ARX model for comparison. The result for the linear model shows its lack of generalization ability for this example. The NARX models show a satisfactory result, where FS-PLS and FS-RR obtain quite the same level of performance. Finally, LS-SVM with a direct subsampling for the computation of the model in dual space, obtains a RMSE level in the test set within the same order of magnitude, but almost twice the one obtained by FS-PLS or FS-RR.

Figure 7.5 shows the residuals of the iterative prediction (simulation mode), where it can be seen that the error remains within a stable zone, with the exception of very few peaks close to the wider zone of the "head of the arrow". In any case, the larger peak represents a 5% absolute error with respect to the level of the output series in that point. The FS-PLS variant of the LS-SVM achieves a root mean squared error (RMSE) of 3.2×10^{-4}, being the best results of the benchmark study.

Technique	Lags	$\text{RMSE}_{\text{test}}$
Linear	30	0.2680
LS-SVM	5	6.2×10^{-4}
FS-OLS	7	6.1×10^{-4}
FS-RR	10	3.3×10^{-4}
FS-PLS	10	3.2×10^{-4}

Table 7.3: *RMSE with the final iterative prediction (simulation mode) on the test data.*

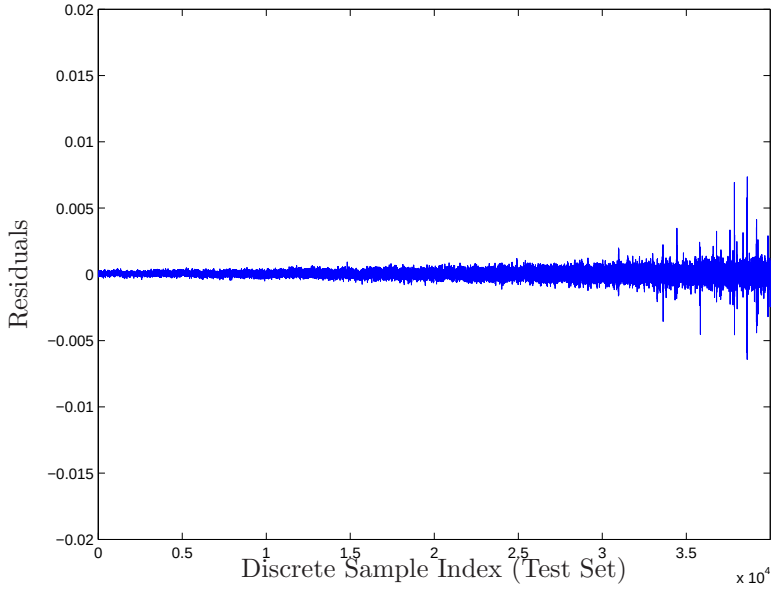

Figure 7.5: *Residuals of the iterative prediction (simulation mode) in the test set. Only few peaks with larger errors are visible.*

7.3 Including Symmetry

The nonlinear component of the system is later known to be a cubic term, which has a symmetry that can be included in the LS-SVM formulation. Remarkably, starting from the polynomial kernel selected from the previous section, and building the equivalent kernel for the case of an odd function (3.9) from it, the results improve substantially with respect to the fully black box model.

Figure 7.6 shows the residuals obtained with standard LS-SVM (top) and symmetric LS-SVM (Bottom) in the simulation exercise. In spite of the very good performance of the black-box model, achieving a root mean squared error (RMSE) of 3.2×10^{-4}, there are still some larger residuals to the end of the sequence, the zone of wider amplitude of the dataset. Imposing symmetry improves the generalization performance on the simulation by reducing the RMSE to 2.8×10^{-4}. Fewer peaks are visible in the residuals obtained with symmetric LS-SVM.

7.4 Using a Partially Linear Model

The full black-box model reached excellent levels of performance using 10 lags of inputs and outputs, obtaining a root mean squared error (RMSE) of 3.2×10^{-4} in simulation mode. Now the objective is to check if the knowledge of the existence of linear regressors can further improve the simulation performance. A partially linear model using $p = q = 10$ is formulated using past and current inputs as linear regressors,

$$\begin{aligned} y_t &= \boldsymbol{\beta}^T [u_t; u_{t-1}; u_{t-2}; \ldots; u_{t-p}] \\ &\quad + \boldsymbol{w}^T \boldsymbol{\varphi}([y_{t-1}; y_{t-2}; \ldots; y_{t-p}]) + e_t \end{aligned}$$

and estimated with PL-LSSVM (6.9). Due to the large sample size, a fixed-size PL-LSSVM in primal space is used. It improves the simulation performance over the full black-box model, as it is shown in Figure 7.7.

Table 7.4 shows a comparison between both models in terms of in-sample accuracy, validation performance, the simulation accuracy and the model complexity. By imposing a linear structure the simulation root mean squared error decreases to 2.7×10^{-4}. Moreover, when considering only the last 10,000 points of the test data, the improvement is more important, as shown

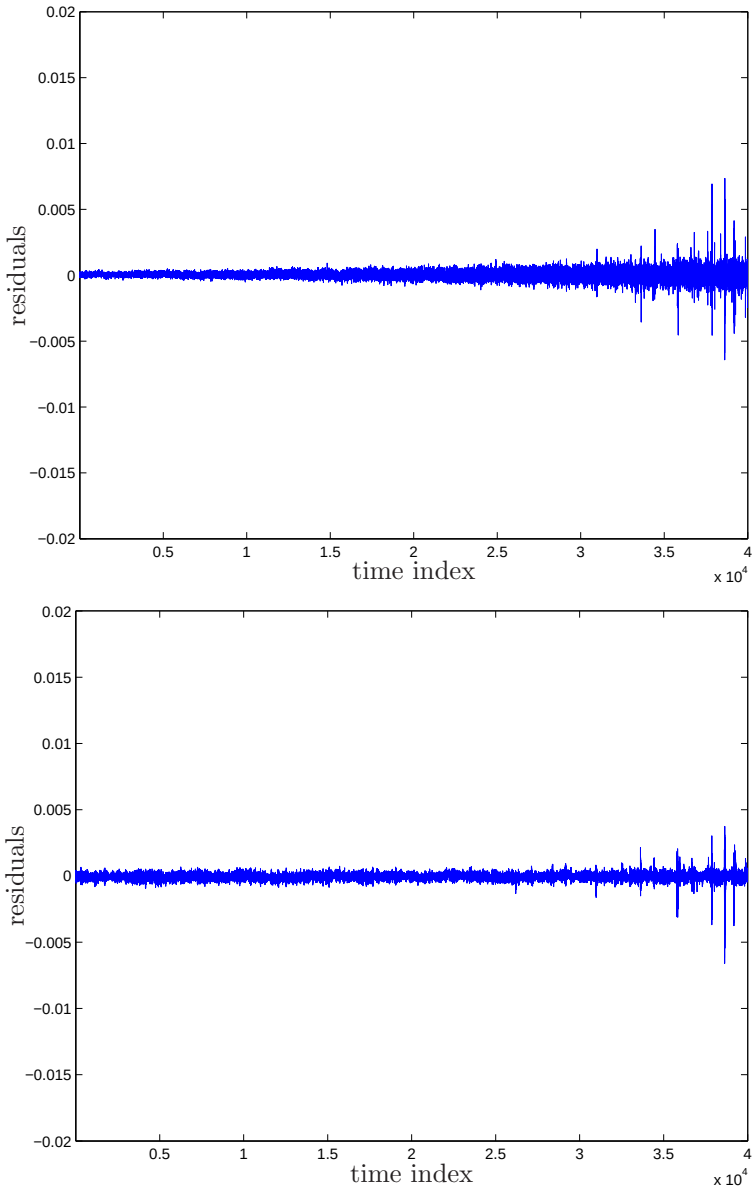

Figure 7.6: *Residuals of the SilverBox simulations on the test set. Standard LS-SVM (Top) and improved results by the symmetric LS-SVM (Bottom).*

	Black-Box model	Partially Linear Model
RMSE_{val}	1.05×10^{-4}	0.45×10^{-4}
$\text{RMSE}_{\text{test}}$	1.70×10^{-4}	0.57×10^{-4}
RMSE_{sim}	3.24×10^{-4}	2.71×10^{-4}
N_{eff}	490	190

Table 7.4: *Performance comparison between the models for the Silverbox data in terms of RMSE for validation, testing and simulation.*

in Table 7.5. Using the full black-box model, the maximum absolute error is 8.1×10^{-3}, which is reduced to 3.7×10^{-3} with the PL-LSSVM. The mean absolute error for the full black-box model is 2.3×10^{-4} and for the partially linear model, 2.02×10^{-4}. The effective number of parameters is reduced from 490 to 190.

Case	Indicators	Black-Box model	Partially Linear Model
Case I	$\max(\lvert e_i \rvert)$	8.1×10^{-3}	3.7×10^{-3}
	$\text{mean}(\lvert e_i \rvert)$	2.30×10^{-4}	2.02×10^{-4}
	$\text{RMSE}(e_i)$	3.24×10^{-4}	2.71×10^{-4}
Case II	$\max(\lvert e_i \rvert)$	8.1×10^{-3}	3.7×10^{-3}
	$\text{mean}(\lvert e_i \rvert)$	3.72×10^{-4}	2.31×10^{-4}
	$\text{RMSE}(e_i)$	5.86×10^{-4}	3.34×10^{-4}

Table 7.5: *Simulation errors for the Silverbox data, over the full test set (Case I) and only for the last 10,000 points of the test set (Case II).*

7.5 Conclusions

The application of the LS-SVM methodology in a large scale nonlinear identification problem implies the challenge of working with a large number of datapoints. In this case, Fixed-Size variants of the LS-SVM are developed to work in primal space using approximations of the nonlinear mapping $\boldsymbol{\varphi}$. These techniques have the advantage that traditional tools available for regression can be applied successfully, and clearly they can deal with large scale problems. In this chapter, we have applied LS-SVM in dual space and 3 variants in primal space (fixed size - ordinary least squares, FS-OLS; fixed size ridge regression, FS-RR; and fixed size partial least squares, FS-PLS)

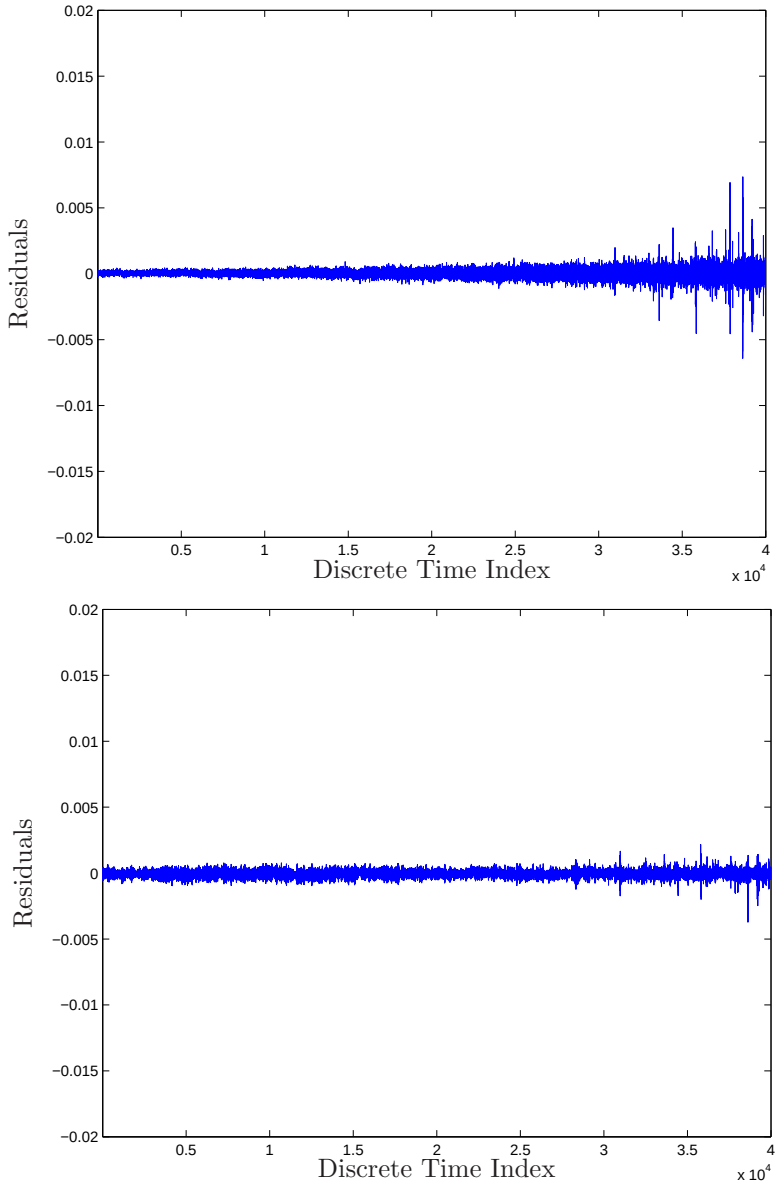

Figure 7.7: *Simulation errors in the test region of the Silverbox data. Full black-box LS-SVM model (Top), PL-LSSVM (Bottom).*

to the identification of a nonlinear dynamical system using the "Silver Box" data.

The results show that methods relying on regularization and active construction of a regression in primal space (FS-RR and FS-PLS) obtain the best performance in the iterative prediction exercise. FS-OLS obtains a lower performance, but still better than the one obtained by the LS-SVM method with a direct sampling. The best performance yields an RMSE on the test set of 3.2×10^{-4}. Computationally, the full estimation methodology can take a few hours, including the selection of hyperparameters. The above results are obtained with models under a suboptimal strategy for model validation and selection. Usually crossvalidation in multiple training/validation sets are done, and in this case we only used one dataset for training and one for validation. Although this is a practical decision, mainly related to the number of datapoints available, results could be improved with a more optimal strategy.

In addition, this chapter shows that it is possible to use prior knowledge to improve over the black-box results. Using a partially linear model with LS-SVM, the methodology is able to successfully identify a model containing a linear part and a nonlinear component, with better performance than a full nonlinear black-box model. The structured model may show a better generalization ability, and a reduced effective number of parameters, than a full nonlinear black-box model. In the same way, incorporating the symmetry of the nonlinear function gives a similar improvement over the initial results.

Part III

Short-Term Load Forecasting

Chapter 8

A Black-Box Approach for Load Forecasting

In the previous chapters, a framework for nonlinear system identification based on modular LS-SVM extensions has been developed. This framework is now applied to the real-life industrial problem of short-term electric load forecasting. This chapter presents the application of black-box NARX models, describing all steps required, from the initial support vector selection, hyperparameter selection, estimation in primal space using a small fraction of the available data to build the nonlinear mapping approximation, and the final assessment of the quality of the model. A comparison is made with a linear ARX model estimated from the same set of explanatory variables, in terms of the forecasting ability of the models for different forecasting horizons. This chapter is structured as follows. An introduction to the problem of short-term load forecasting is given in Section 8.1. The description of the available data, the definition of the model structure and the estimation method are described on Section 8.2. The empirical results are discussed in detail in Section 8.3.

8.1 Problem Description

This section provides an introduction to the short-term load forecasting context. The practical importance for electric energy generators, grid operators, suppliers and other market players is described, and an assessment of existing forecasting techniques is given.

8.1.1 The Short-Term Load Forecasting Problem

The problem of short-term load forecasting (STLF) is an important area of quantitative research in power systems [29, 63, 78, 97]. STLF refers to the prediction of the system load over an interval ranging from an hour (or fraction) to one week. From the power generation perspective, accurate tracking of the load by the system generation at all times is a basic requirement in the operation of power systems [47, 84], which must be accomplished for different time intervals. It is a task that is used on a daily basis on every major dispatch center or by grid managers. For example, for the time scale of hours, variations of the load can occur which would require the startup or shutdown of entire generating units. It is therefore important not only to be able to predict the hourly load in general, but also the daily peak system load in particular. Electricity cannot be efficiently stored in large quantities, meaning that the amount generated at any given time always has to cover all the demands from the final consumers, including grid losses. Forecasts of the load are used to decide whether extra generation has to be provided by increasing the output of on-line generators or by committing one or more extra units. Similarly, forecasts are also used to decide whether an already running generation unit should be decreased in output or even switched off. Moreover, the flow along the transmission lines is affected by the different generation profiles, possibly leading to congestion problems. On the other hand, the liberalization of the electric energy markets has led to the development of energy exchanges, where consumers, generators and traders can interact leading to price settings. In this respect also forecasts are extremely important, as the liberalization of the electricity sector has given a new dimension to the problem of STLF [14, 17]. Large time series, provided by the Belgian Transmission System Operator (TSO) ELIA, are used in this article as examples to illustrate the importance and possibilities of the implementation of nonlinear system identification techniques for short-term load forecasting. The available time series contains

hourly load values taken from different substations within the Belgian grid. Such substations correspond to the off-take points used by local distribution companies. The voltage is converted from high-voltage, usually above 70 kV, to the required level on each substation [87].

In general, building a model for load forecasting is not straightforward, due to the presence of seasonal patterns in different levels. There is a winter-summer pattern, a weekly pattern and an intra-daily pattern. Figure 8.1 shows an example of a load series in a week, at hourly values starting at 00:00 h on Monday, until 24:00 h on Sunday. These different patterns also interact with other external variables that affect the load, the weather fluctuations being one of the most importants. When the weather is cold, there is a requirement for heating which translates in an increase of the energy demand. Hot days in summer trigger the use of air conditioning equipment, also increasing the demand. On the other hand, the load on a Monday looks like the previous Monday, but a Monday in winter is different from a Monday in summer, as shown on Figure 8.2. The same can be observed for weekends. However, special days (for example, May 1st, Easter, Christmas) can show a different behavior. All these effects can combine with each other, and for example the effect of the weather on a winter Monday is different from the effect of the weather on a summer Friday. The effect of weather in the load is nonlinear, which is one of the main reasons for using nonlinear models for this problem, as seen on Figure 8.3. In addition, for the purposes of long-term and mid-term planning, year-to-year comparisons and scenario analysis, it is important to have interpretable models. A model has to be able to tell how much of the peak was due to the weather conditions of that particular day, so it can be corrected towards a normal meteorological year. Other types of analysis can be done by e.g. comparing the consumption of different regions or identifying customer profiles.

8.1.2 Existing Methodologies

For the problem of STLF, the main goal is to generate a model that can capture all the dynamics and interactions between possible explanatory variables of the load. For this task, it is found in the literature that there is a broad consensus about possible explanatory variables: past values of the load, weather information, calendar information, and possibly some past-errors correction mechanisms. In the literature, it is often found that some local models of the load are used to produce short-term

forecasts; the local models are selected in order to isolate a seasonal pattern (working only with winter, summer, evenings, working-days, etc). By following a seasonal-modelling approach, it is possible to incorporate a priori information by appropriately choosing the model structure. Then, a priori information concerning the seasonalities at several levels (daily, weekly, yearly, etc.) can be included directly in the model. Within the time series literature, it is known that seasonality can be modelled in different ways. The simplest approach is to assume deterministic seasonality that can be represented by including binary or dummy variables in the model. More complex approaches include the assumption of stochastic seasonality, in the framework of Box-Jenkins seasonal ARIMA models [13, 91], leading to testing for seasonal unit roots in time series analysis [64]; the use of nonparametric models with seasonal components [139]; or, more recently, the application of seasonal-varying parameters in an autoregression [29, 42]. This has led to the development of a wide range of models based on different techniques. In recent literature, some interesting examples are related to traditional time series analysis [8, 63, 97], and neural networks applications [41, 71, 72, 89, 110].

8.2 Modeling Strategy

In this section, the modeling strategy is described in terms of dataset definition, model structure and estimation procedure.

8.2.1 Data Definition

The dataset consists of several time series, each containing hourly load values from different HV-LV substations within the Belgian grid for a period of approximately 5 years (from January 1998 until September 2002). The load series differ in their behavior as they represent different types of underlying customers (residential, business, industrial, etc.). A linear regression containing only a linear trend is estimated for each load series, to remove any growth trend present in the sample. Finally, the series are normalized using the mean and standard deviation of the sample.

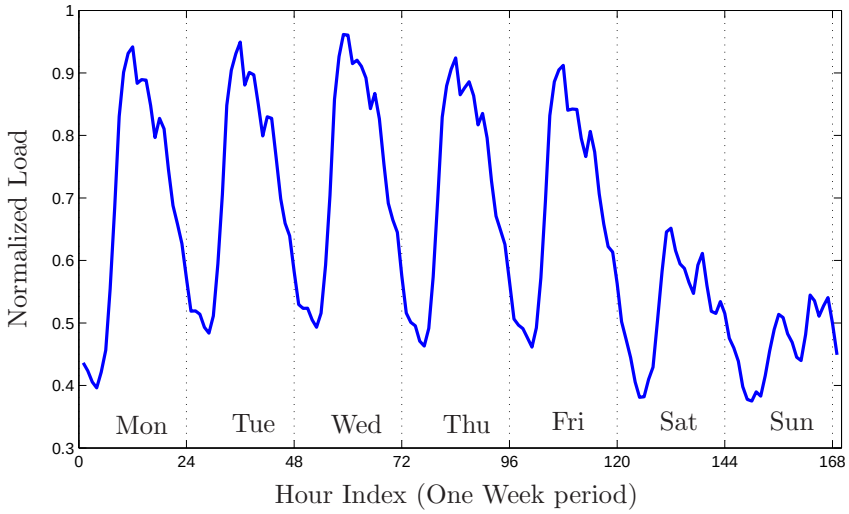

Figure 8.1: *Example of a Load series within a week. Daily cycles are visible, as well as the weekend effects. Also visible are the intra-day phenomena, such as peaks (morning, noon, evening) and valleys (night hours).*

8.2.2 Using Nonlinear Black-Box Models

(N)ARX type of models are considered in the context of this chapter. The load at a given hour is explained by the evolution of the load in previous hours, and by the effect of exogenous variables keeping track of the different seasonal patterns. For example, the dummy (binary) variable $W_d \in \mathbb{R}^7$ is a vector of zeros with a "1" in the position of the day d in the week, e.g. Monday ($W_d = [1; 0; \cdots; 0]$), Tuesday ($W_d = [0; 1; 0; \cdots; 0]$), etc. Similarly, the variable $M_d \in \mathbb{R}^{12}$ is defined as a vector of zeros with a "1" in the position of the month to which the day d belongs[1]. In addition, temperature variables are included in order to capture the effect due to the weather conditions. The hourly temperature variable T_h is the observed local temperature at hour h at a reference location (Ukkel) in Belgium. From T_h, 3 new variables are built to capture the effect of cooling and heating

[1]In case of a linear ARX model, to avoid exact collinearity between all the calendar variables and the constant terms in the original system (9.2) only 6 of the W_d components and only 11 of the M_d components are incorporated in the model. This is a standard implementation of dummy variables in any econometric estimation procedure [67].

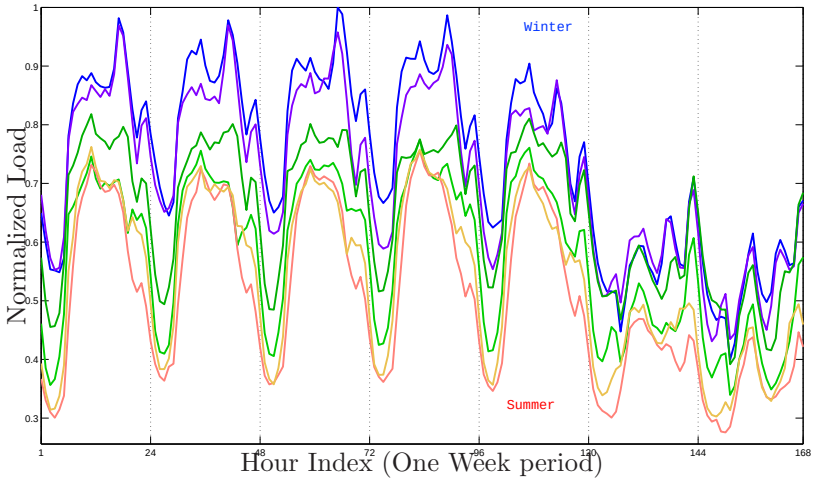

Figure 8.2: *Comparison of a weekly profile over the year. The load in Winter (dark-top curve) is different from the load in Summer (bottom-light curve), both being different from a profile from Spring or Autumn (intermediate curves). Notice the pronounced evening peaks that only occur in Winter for this substation.*

requirements [25] in the load. The variable $\mathrm{CR}_h = \max(T_h - 20^o, 0)$ is defined for capturing the cooling requirement, if the ambient temperature is above 20^oC. Similarly, heating and extra-heating variables are defined using $\mathrm{HR}_h = \max(16.5^o - T_h, 0)$ and $\mathrm{XHR}_h = \max(5.0^o - T_h, 0)$, respectively, with the temperature thresholds taken from standard techniques within the energy industry. Therefore, T_h has been expanded into a vector $\boldsymbol{v}_h = [\mathrm{CR}_h, \mathrm{HR}_h, \mathrm{XHR}_h]$.

The model formulation to be used contains the following explanatory variables:

- An autoregressive part of 48 lagged load values (i.e. the last 2 days)
- Temperature-related variables measuring the effect of temperature on cooling and heating requirements (3 variables)
- Calendar information in the form of dummy variables for month of the year, day of the week and hour of the day (43 variables)

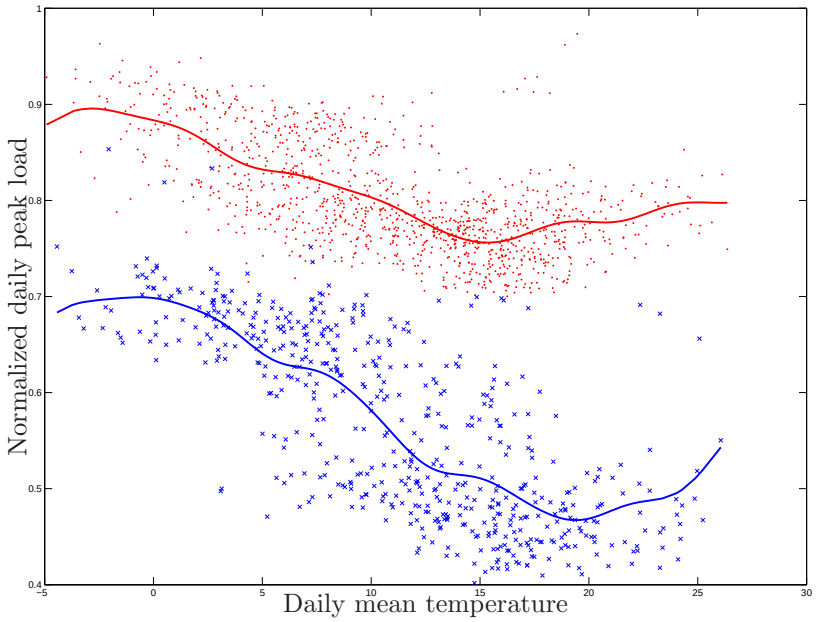

Figure 8.3: *Nonlinear relation between temperature and load. One of the reasons to use nonlinear models is due to the relation between the ambient temperature and the observed load. Cold days trigger more energy consumption, as well as very warm days. The daily peak load is plotted against the daily mean temperature, for working days ('.') and weekends ('x'). The nonlinear relation is captured by a LS-SVM regression, represented by thick lines on each case. A forecasting model needs to be able to cope with this nonlinear effect.*

This leads to a set of 94 (48+3+43) explanatory variables to be included in the regression vector $\boldsymbol{z}_t$ of the black-box NARX model

$$y_t = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}_t) + b \tag{8.1}$$

where y_t is the load at time t, b is a constant (bias) term, and $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{N_h}$ is the feature map.

A sample of 1500 days (36,000 hours) is considered for training the models, estimated (due to the large number of datapoints) in primal space

(6.21) using the Nyström method described on Chapter 2. Tuning of the RBF kernel hyperparameter σ and the regularization parameter γ is performed by 10 fold cross validation in the training sample, selecting those hyperparameters that minimize the cross validation mean squared error (MSE).

To illustrate the effect of M, the number of initial support vectors from which the nonlinear approximation (2.22) is computed, on the forecasting performance of the estimated model, the methodology is tested for sizes of $M = 200, 400, 600, 800$ and 1000 support vectors. Each time, the support vectors are selected using the quadratic Renyi entropy criterion. It is important to stress out that between 0.5% and 3% of the dataset is used to build the nonlinear mapping for the entire sample, translating in a sparse representation of the nonlinear mapping. Having a sparse representation makes the model less prone to overfitting, which has been an important issue in recent literature [59]. Values of M larger than 1000 are possible, as the only constraint in this approach is the computational time depending on the resources at hand.

The performance of this fixed-size LS-SVM black-box model is compared to the performance of a linear ARX model estimated with the same initial set of variables, i.e., using the same regression vector $\boldsymbol{z}_t$. In addition, the comparison is extended to include the performance of a standard LS-SVM in dual form, estimated using only the last 1000 datapoints of the sample. In this way, it is possible to compare the difference in performance between 2 nonlinear models in the following two cases: when the full sample is taken into account (fixed-size LS-SVM) or only when the most recent 1000 hours (last 42 days) are considered.

The forecasting performance is assessed as follows. The simplest scheme is to forecast the first out-of-sample load value using all information available, then wait one hour until the true value of this forecast has been observed, and then forecast the next value again using all available information (one-hour-ahead prediction). However, planning engineers require forecasts with a longer time horizon, at least a full day in advance. In this case, it is required to predict the first out-of-sample value using the full working sample, then predict the second value out-of-sample using this first prediction, and so on (iterative simulation). In practice, it is reasonable to stop this iterative process after 24 hours and update the information with actual observations. The methods are compared by looking at their test set performance, defined on a test data not used during training/estimation consisting of the block

of 15 days after the last training point. The performance is assessed via the Mean Squared Error (MSE) for the one-step-ahead prediction and the 24-hours-ahead-simulation with updates at 00:00 hrs. of each day. In these forecasting exercises, the external variables are assumed to be known. This is not a problem for the calendar variables, although external weather forecasts [118] should be used instead of actual temperature values. Nevertheless, using actual values for temperature as inputs for the load forecasting helps to assess the model performance without additional error sources. On the other hand, using different temperature values leads to simulation exercises, where the aim is to look at what would be the load if the temperature pattern changes.

8.3 Empirical Results

In this section the results of the fixed-size LS-SVM methodology applied to the load modelling problem are discussed, regarding the training procedure, selection of support vectors and out of sample performance.

8.3.1 Cross-Validation Performance

The above procedure is applied for $M = 200, 400, 600, 800$ and 1000. Training using 10 fold crossvalidation is performed for each case, looking for an optimal value of the hyperparameter σ in the RBF kernel. Figure 8.4 shows the evolution of the MSE in the 10 fold crossvalidation training procedure for the cases of $M = 200$ and $M = 400$ in one of the load series, where it can be seen that the optimal value is $\sigma = 2.01$. For the cases $M = 600$, $M = 800$ and $M = 1000$ the crossvalidation process using only the selected σ is performed. The results for the computed MSE in a crossvalidation basis, and the equivalent result for the linear model, are shown in Table 8.1 using the optimal σ.

8.3.2 Support Vector Selection

The initial set of M support vectors has been selected by maximizing the quadratic Renyi entropy (2.24). In this way, it is possible to obtain a selection of those M points that converge to a maximum value of the

Estimation	Mean Squared Error (CV)
Linear	0.043
M=200	0.032
M=400	0.022
M=600	0.017
M=800	0.016
M=1000	0.015

Table 8.1: *Performance of the Fixed-Size LS-SVM models where the nonlinear mapping approximation is built with M support vectors, on a crossvalidation basis using the optimal σ.*

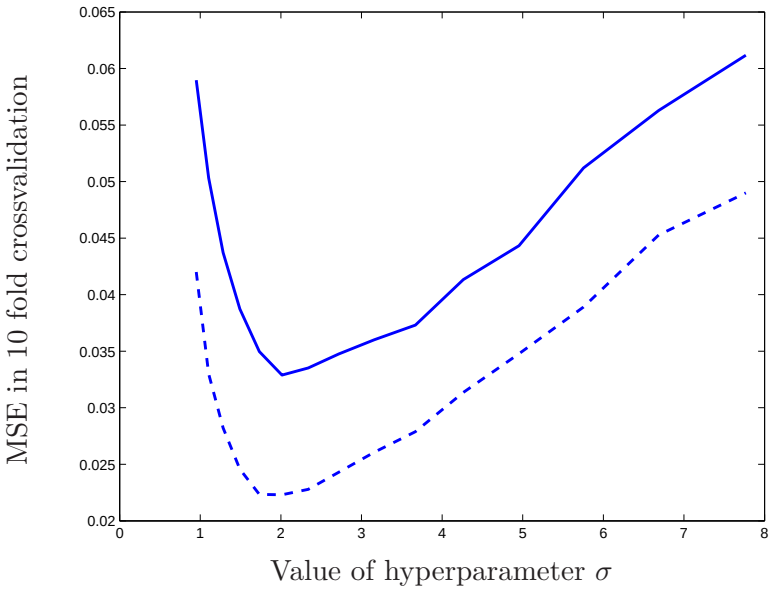

Figure 8.4: *Performance evolution in the training procedure. The lines show the evolution of the MSE in a 10 fold crossvalidation for the cases $M = 200$ (full line) and $M = 400$ (dashed line). The optimal value for the σ hyperparameter is 2.01.*

quadratic entropy. Figure 8.5 shows the evolution of the entropy value within this iterative process (for a selected load series), for the case of $M = 400$. Figure 8.6 shows the position (time index) of the first element of the selected support vectors for the case of $M = 400$. It is interesting to see how the selected support vectors are those for which the output series is located in the regions of high load values (Winter), some in the lower values (Summer) and almost none of them in Spring or Autumn.

8.3.3 Effect of Selection Method

It is interesting to verify the effect of the support vector selection method into the final accuracy of the model. Maximizing the quadratic entropy is a fast procedure, as it only requires the computation of a small kernel matrix of dimension M. However, it is interesting to check what would happen if a random selection of support vectors is made. Consider the following 2 cases. In the first case (Case I), a random sample of M support vectors is used as the starting point for the iterative procedure of maximization of the quadratic Renyi entropy. After the process has converged, the final selection of support vectors is used to build the approximation of the nonlinear mapping $\boldsymbol{\varphi}$ using the Nyström techniques. In the second case (Case II), the random sample of M support vectors is used *directly* to build the nonlinear approximation, thus no entropy is maximized. Both models are estimated in primal space, and the forecasting performance on a test set is compared. For this purpose, 20 different experiments are computed. Table 8.2 and Figure 8.7 show the comparison of the results after 20 random initial selections, for the case where the model is estimated after doing a quadratic entropy selection (Case I), or estimated directly (Case II). In all tests it has been used $M = 200$.

The existence of the standard deviation in Case I accounts for the fact that the convergence of the entropy selection is not unique, especially for a selection of 200 points out of 36,000 possible samples. However, starting from different random selections, the entropy-based selection yields lower dispersion in the forecasting errors. For this dataset, and after 20 repetitions, the average MSE for both models are quite similar, but there is no guarantee that the random-selection will show this performance for a more complex dataset.

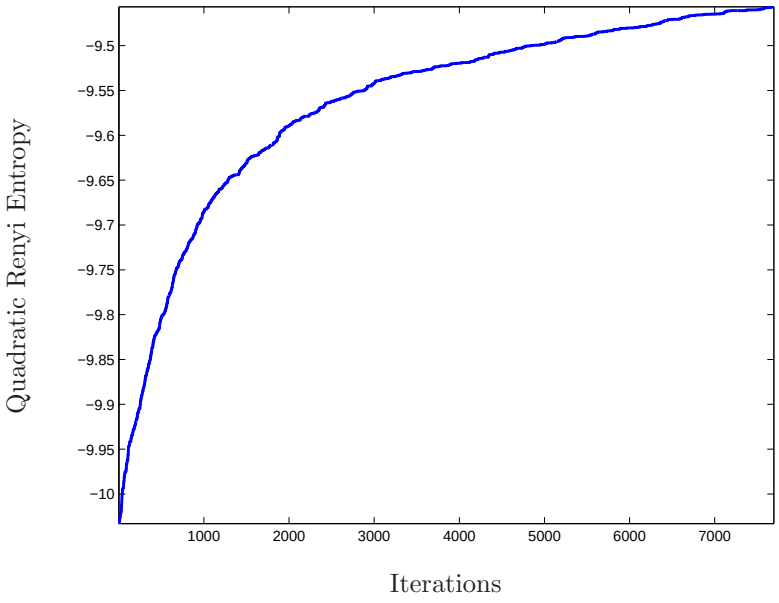

Figure 8.5: *Evolution of the quadratic Renyi entropy within the iterative search for support vectors with* $M = 400$.

Support Vector Selection Method	Mean Squared Error (MSE)	
	Average	Standard Deviation
Entropy-based Selection (Case I)	0.0311	0.0016
Random-based Selection (Case II)	0.0317	0.0025

Table 8.2: *Comparison of the average and standard deviation of the MSE for a test set performance using* $M = 200$ *over 20 randomizations. Case I refers to the random selection of support vectors and immediate estimation of the model. Case II starts from the same random selection, performs quadratic-based selection using the random sample as starting point, and then the model is estimated.*

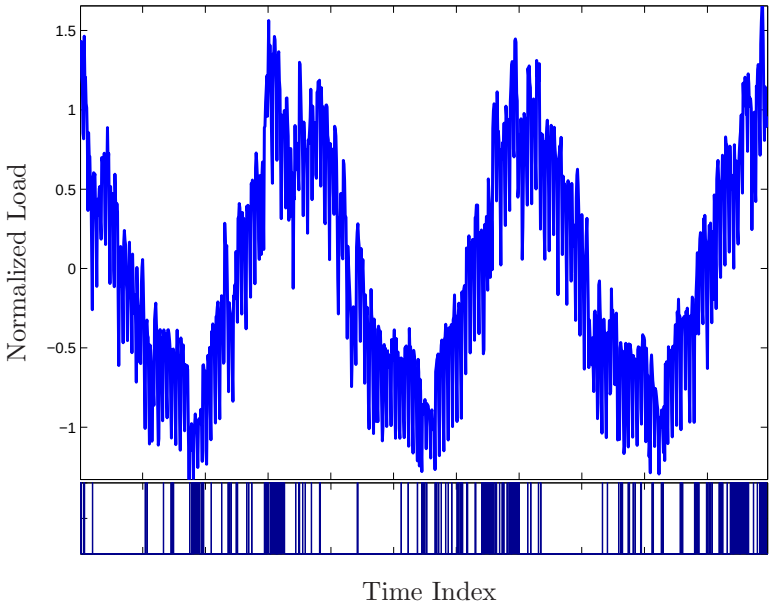

Figure 8.6: *Normalized load from Series 1 used as training sample, shown here only as daily averages rather than hourly values. The position of the selected support vectors corresponding to the load sample output is represented by dark bars at the bottom, showing the time index position of the first element of the support vector.*

8.3.4 Test Set Performance

The forecasting ability of the models is compared on a test set consisting of the next 15 days after the last training point. The performance is assessed over 2 forecasting modes: one-hour-ahead prediction, and 24-hours-ahead-simulation with updates at 00:00 hrs. of each day. The performance of the predictions obtained from any given model can be assessed by using both MSE and MAPE (Mean Absolute Percentage Error).

The MSE is typically used within the general context of applied modeling. Here the MAPE is also used as it is common practice in the particular context of the STLF. Therefore, in this case study the performance of the models is assessed using both indicators. Three models are estimated for each load series: the fixed-size LS-SVM (FS-LSSVM) estimated using the entire sample, the standard LS-SVM in dual version estimated with the last

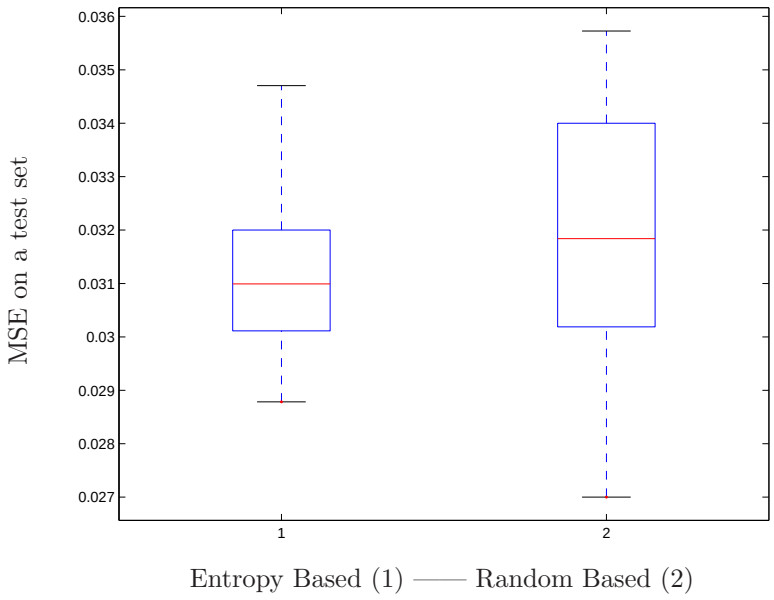

Figure 8.7: *Box-plot of the MSE in a test set for models estimated with entropy-based (1) and random (2) selection of support vectors. Results for 20 repetitions.*

1000 datapoints of the training sample, and a linear model estimated with the same variables as the FS-LSSVM.

The fixed-size LS-SVM models are computed using $M = 1000$ initial support vectors. The difference across the forecasting ability over the series is due to the different behavior of each particular load series. Tables 8.3 and 8.4 show the comparison between the models performance for the different forecasting modes over the 10 load series. Clearly, the fixed-size LS-SVM improves over the traditional LS-SVM, mostly as it uses more information by including the entire datasample available, rather than just using the last 1000 datapoints. In the context of load-forecasting, the existence of important seasonal variations makes it important to include as many datapoints as possible into the model. On the other hand, the linear model shows good performance in some series, but it is always outperformed by the fixed-size LS-SVM. Linear models for load forecasting have to be designed in more detail to improve its performance, through the explicit incorporation of their seasonal variations across days and weeks into the model (e.g. periodic

linear autoregressions, as it will be seen on Chapter 9). The nonlinear model requires less effort from the user in the definition of the model, and the whole procedure can be programmed to be done automatically.

The comparison between the forecasts obtained with the fixed-size LS-SVM and the linear model are shown on Figures 8.8, 8.9 and 8.10 for Series 3, 4 and 9, respectively. In each figure, the top panels show the performance using one-hour-ahead forecasts. The bottom panels show the comparison using 24-hours-ahead simulation. Each plot shows the first 7 days of the test set, starting with 00:00 hrs on Monday. Clearly, the fixed-size LS-SVM model provides better forecasts, particularly for the case of 24-hours-ahead prediction. It is also interesting to notice the different daily profile of each load series.

8.4 Conclusions

This chapter illustrated the application of a black-box NARX model to the short-term load forecasting problem, estimated using a large-scale nonlinear regression technique. It has been shown that it is possible to build a large scale nonlinear regression model, using the fixed-size LS-SVM, from a dataset consisting of $N = 36,000$ datapoints. This is done by selecting an initial subsample of size $M \ll N$, providing a sparse representation of the nonlinear mapping. The results show that the nonlinear regressions in primal space improve their accuracy with larger values of M. The maximum value of M to be used depends on the computational resources at hand, and also on the underlying distributional properties of the dataset. In this context, it is shown that quadratic entropy active selection of support vectors leads to performances with a lower dispersion as those obtained by random selection of support vectors.

The forecasting performance, assessed for 10 different load series, is very satisfactory. The MSE levels are below 3% in most cases. Not only the model estimated with fixed-size LS-SVM produces better results than a linear model estimated with the same variables, but also it produces better results than a standard LS-SVM in dual space estimated using only the last 1,000 datapoints. This shows that it is important to consider as much datapoints as possible into the modeling task. Furthermore, the good performance of the fixed-size LS-SVM is obtained based on a subsample of $M = 1000$ initial support vectors, representing less than 3% of the available sample.

Series	Mode	Performance	LS-SVM	FS-LSSVM	Linear
Series 1	1-h-ahead	MSE	2.2%	0.6%	1.4%
		MAPE	2.8%	1.5%	2.5%
	24-h-ahead	MSE	5.0%	2.7%	9.5%
		MAPE	4.3%	3.1%	5.9%
Series 2	1-h-ahead	MSE	3.4%	2.3%	3.0%
		MAPE	4.3%	3.4%	3.9%
	24-h-ahead	MSE	20.2%	11.5%	11.9%
		MAPE	10.6%	7.4%	7.9%
Series 3	1-h-ahead	MSE	9.7%	6.7%	10.2%
		MAPE	29.4%	17.7%	24.9%
	24-h-ahead	MSE	15.1%	9.4%	15.0%
		MAPE	30.1%	23.1%	29.7%
Series 4	1-h-ahead	MSE	4.9%	4.0%	7.4%
		MAPE	12.6%	10.5%	16.2%
	24-h-ahead	MSE	10.1%	6.0%	14.7%
		MAPE	20.7%	14.5%	22.3%
Series 5	1-h-ahead	MSE	2.2%	0.9%	1.7%
		MAPE	2.6%	1.7%	2.2%
	24-h-ahead	MSE	9.0%	3.8%	6.7%
		MAPE	5.5%	3.4%	4.4%

Table 8.3: *Model performance on the test set for different forecasting modes, for series 1-5.*

Further research on a more dedicated definition of the initial input variables (e.g. incorporation of external variables to reflect industrial activity, use of explicit seasonal information, etc.) should lead to further improvements.

Series	Mode	Performance	LS-SVM	FS-LSSVM	Linear
Series 6	1-h-ahead	MSE	0.8%	0.3%	1.1%
		MAPE	2.3%	1.4%	2.2%
	24-h-ahead	MSE	3.9%	2.6%	7.5%
		MAPE	5.1%	4.4%	7.1%
Series 7	1-h-ahead	MSE	2.6%	1.6%	3.0%
		MAPE	2.9%	2.2%	3.1%
	24-h-ahead	MSE	5.7%	3.8%	6.8%
		MAPE	4.5%	3.5%	4.7%
Series 8	1-h-ahead	MSE	2.4%	1.5%	2.2%
		MAPE	3.0%	2.4%	2.8%
	24-h-ahead	MSE	9.8%	5.3%	7.7%
		MAPE	7.3%	4.4%	5.3%
Series 9	1-h-ahead	MSE	0.9%	0.5%	1.3%
		MAPE	1.8%	1.3%	2.0%
	24-h-ahead	MSE	3.2%	2.1%	6.9%
		MAPE	3.4%	2.8%	5.3%
Series 10	1-h-ahead	MSE	2.8%	2.3%	3.5%
		MAPE	5.7%	4.9%	6.0%
	24-h-ahead	MSE	9.9%	8.2%	12.7%
		MAPE	11.0%	10.9%	13.4%

Table 8.4: *Model performance on the test set for different forecasting modes, for series 6-10.*

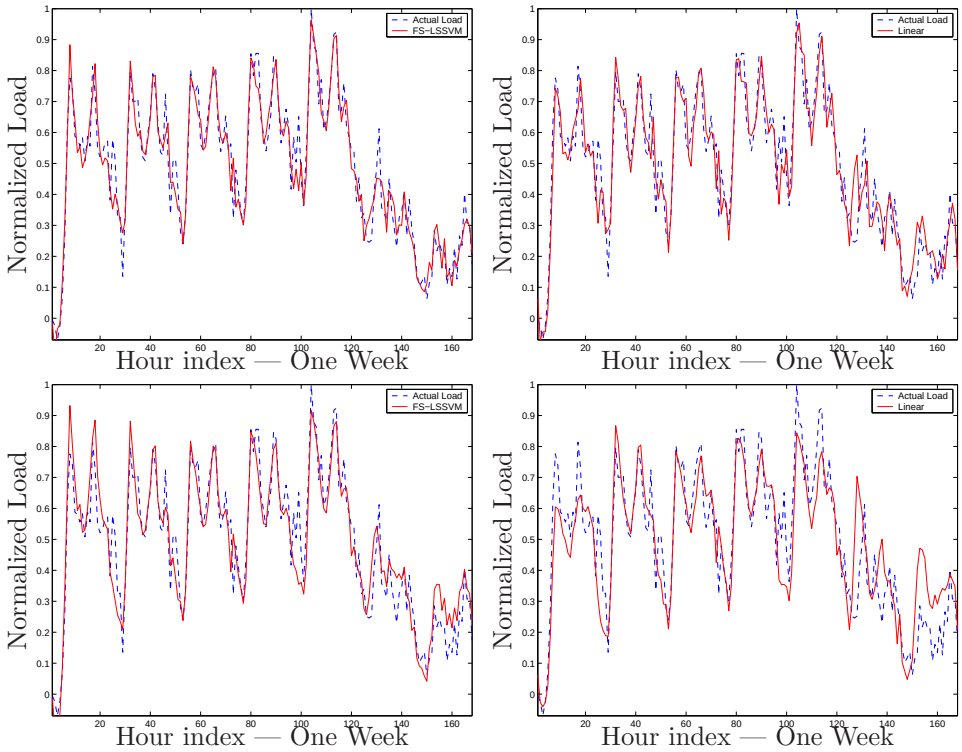

Figure 8.8: *Forecasts comparison. FS-LSSVM and Linear one-hour-ahead predictions (Top-left and Top-right, respectively). FS-LSSVM and Linear 24-hours-ahead predictions (Bottom-left and Bottom-right, respectively), for a full week (Series 3).*

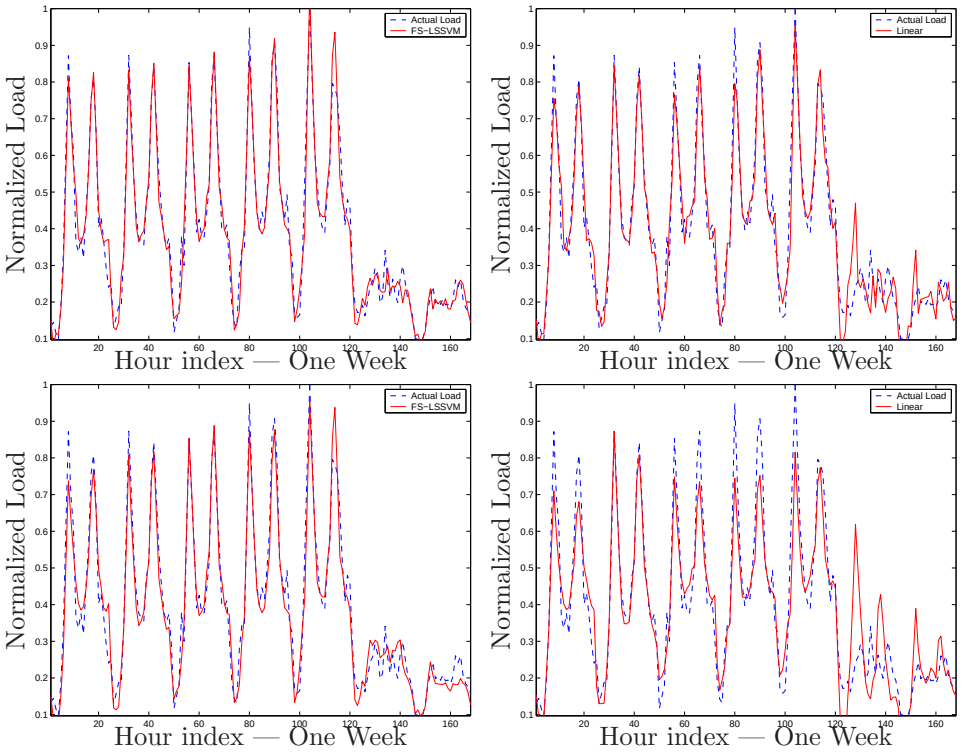

Figure 8.9: *Forecasts comparison. FS-LSSVM and Linear one-hour-ahead predictions (Top-left and Top-right, respectively). FS-LSSVM and Linear 24-hours-ahead predictions (Bottom-left and Bottom-right, respectively), for a full week (Series 4).*

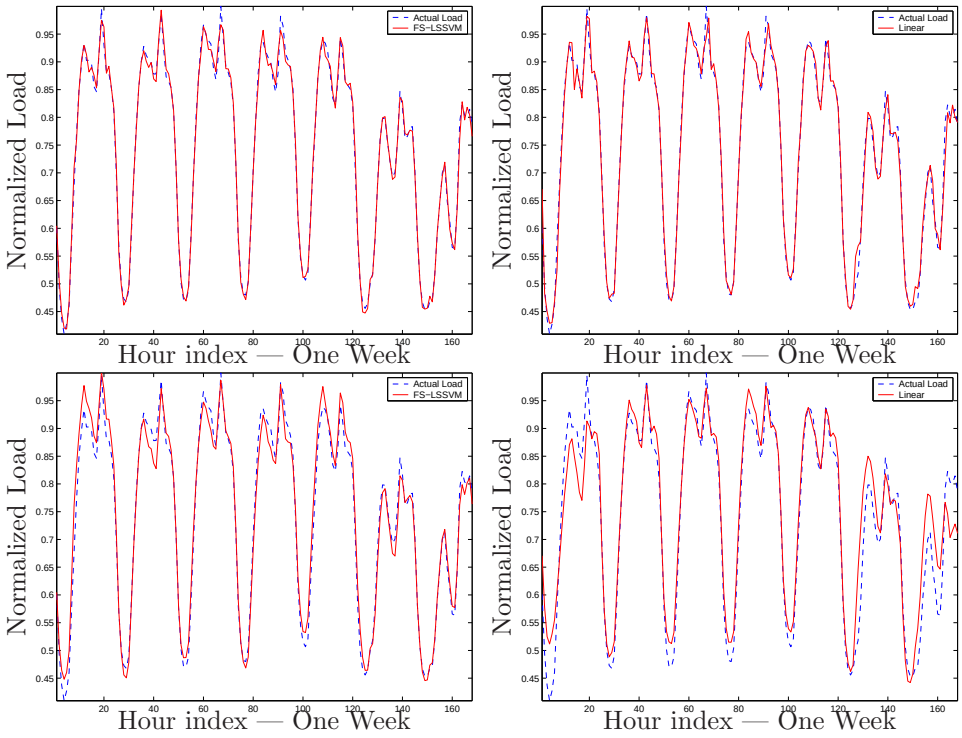

Figure 8.10: *Forecasts comparison. FS-LSSVM and Linear one-hour-ahead predictions (Top-left and Top-right, respectively). FS-LSSVM and Linear 24-hours-ahead predictions (Bottom-left and Bottom-right, respectively), for a full week (Series 9).*

Chapter 9

Load Forecasting with Structured Models

The implementation of structured models for the problem of short-term load forecasting is developed in this chapter. The models are extended to include autocorrelated residuals, and/or linear parametric parts, thus using the AR-NARX, PL-NARX and AR-PL-NARX model structures. In order to make an assessment of the forecasting ability as strict as possible, these structured nonlinear models are estimated using different load series, and their performance is compared with that of a highly structured linear model also developed within the context of this work. The structured linear model provides a strong benchmark for the nonlinear models. In this context, the goal of this chapter is to show the improvement of using structured over unstructured models, and to provide a methodological basis for the use of the structured nonlinear models developed for the problem of STLF. In addition, the use of partially linear structures leads to a set of identified parameters which may yield a direct interpretation, as in the case of linear models, for a set of variables of interest. This chapter is designed as follows. Section 9.1 presents the structured linear model formulation to be used. The nonlinear model formulations to be considered are described on Section 9.2. The empirical methodology and results are reported in Sections 9.3 and 9.4, respectively.

9.1 Structured Linear Models

This section describes the structured linear model formulation based on Periodic Autoregressions.

9.1.1 Periodic Autoregressions

One of the recent approaches for the seasonal modeling of time series is related to the so-called Periodic Autoregressions (PAR) [42, 85, 120]. In simple terms, an autoregression is said to be periodic when the parameters are allowed to vary across seasons. Consider the case of a univariate time series y_t, $t = 1, \cdots, N$, (in this case, the hourly load measurements) available for a sample of $N_d = N/24$ days, corresponding to the N hours. A periodic autoregressive model of order p (PAR(p)) can be written as

$$y_t = C_s + \theta_{s,1} y_{t-1} + \theta_{s,2} y_{t-2} + \cdots + \theta_{s,p} y_{t-p} + \varepsilon_{s,t} \tag{9.1}$$

where C_s is a seasonally varying intercept term and the $\theta_{s,i}$ are the autoregressive parameters up to the order p, varying across the N_s seasons ($s = 1, 2, \cdots, N_s$). The choice of N_s depends on the frequency of the data and the seasonal pattern under study. The error term $\varepsilon_{t,s}$ can be a standard white noise with zero mean and variance σ, or it can be allowed to have a variance σ_s corresponding to seasonal heteroskedasticity. Equation (9.1) gives rise to a system of N_s equations that can be estimated using Ordinary Least Squares (OLS).

9.1.2 Model Formulation

For the model implementation [29,49], the monthly and weekly seasonals are modelled by their corresponding dummy variables as defined in Chapter 8, and the intra-daily seasonal pattern is assumed to be captured by the PAR parameters. In other words, $N_s = 24$ is the number of different "seasons" to be identified using the PAR model. Denote by $y_{h,d}$ the value of the load measured in hour h of day d, with $h = 1, 2, \cdots, 24$ and $d = 1, 2, \cdots, N_d$. A formulation is built where the hourly load $y_{h,d}$ is a function of the last 48 hourly values. The parameter p of the PAR(p) is, therefore, $p = 48$. This value is defined by trying first $p = 24$, $p = 36$ and finally $p = 48$ in order to obtain a satisfactory model performance and, at the same time,

keeping model parsimony. The PAR(48) model applied to the hourly load forecasting problem defines the following set of equations:

$$\begin{aligned} y_{1,d} &= C_1 + \theta_{1,1} y_{24,d-1} + \theta_{1,2} y_{23,d-1} + \cdots + \varepsilon_{1,d} \\ y_{2,d} &= C_2 + \theta_{2,1} y_{1,d} + \theta_{2,2} y_{24,d-1} + \cdots + \varepsilon_{2,d} \\ y_{3,d} &= C_3 + \theta_{3,1} y_{2,d} + \theta_{3,2} y_{1,d} + \cdots + \varepsilon_{3,d} \\ &\vdots \\ y_{24,d} &= C_{24} + \theta_{24,1} y_{23,d} + \theta_{24,2} y_{22,d} + \cdots + \varepsilon_{24,d}. \end{aligned} \tag{9.2}$$

This basic PAR template consists of $24 \times 49 = 1176$ parameters. This template is further extended to include exogenous variables to account for temperature effects as well as monthly and weekly seasonal variations, as described in Chapter 8, by the variables $\boldsymbol{v}, M_d, W_d$.

With the inclusion of the exogenous variables (6 for the week calendar, 11 for the month calendar, and 3 for the temperature-related variables), the total number of coefficients to be estimated using a PAR(48) with $N_s = 24$ is $24 \times (49 + 6 + 11 + 3) = 1656$. The augmented system (9.3) is estimated individually for each one of the available time series, using OLS with $t-$tests of significance to keep only those coefficients statistically different from zero. By using the same model template for all substations in the electricity grid makes it possible to perform all kinds of comparisons in terms of their parameter estimates, accuracy obtained, etc.

$$\begin{aligned} y_{1,d} &= C_1 + \theta_{1,1} y_{24,d-1} + \theta_{1,2} y_{23,d-1} + \cdots + \theta_{1,48} y_{1,d-2} + \\ &\quad + \boldsymbol{\alpha}_1^T W_d + \boldsymbol{\beta}_1^T M_d + \boldsymbol{\gamma}_{1,d}^T \boldsymbol{v}_{1,d} + \varepsilon_{1,d} \\ y_{2,d} &= C_2 + \theta_{2,1} y_{1,d} + \theta_{2,2} y_{24,d-1} + \cdots + \theta_{2,48} y_{2,d-2} + \\ &\quad + \boldsymbol{\alpha}_2^T W_d + \boldsymbol{\beta}_2^T M_d + \boldsymbol{\gamma}_{2,d}^T \boldsymbol{v}_{2,d} + \varepsilon_{2,d} \\ y_{3,d} &= C_3 + \theta_{3,1} y_{2,d} + \theta_{3,2} y_{1,d} + \cdots + \theta_{3,48} y_{3,d-2} + \\ &\quad + \boldsymbol{\alpha}_3^T W_d + \boldsymbol{\beta}_3^T M_d + \boldsymbol{\gamma}_{3,d}^T \boldsymbol{v}_{3,d} + \varepsilon_{3,d} \\ &\vdots \\ y_{24,d} &= C_{24} + \theta_{24,1} y_{23,d} + \theta_{24,2} y_{22,d} + \cdots + \theta_{24,48} y_{24,d-2} + \\ &\quad + \boldsymbol{\alpha}_{24}^T W_d + \boldsymbol{\beta}_{24}^T M_d + \boldsymbol{\gamma}_{24,d}^T \boldsymbol{v}_{24,d} + \varepsilon_{24,d} \end{aligned} \tag{9.3}$$

9.2 Structured Nonlinear Models

9.2.1 AR-NARX Model

The black-box NARX model estimated in Chapter 8 is extended by imposing autocorrelation on the residuals,

$$\begin{cases} y_t = g(\boldsymbol{z}_t) + e_t \\ e_t = \rho e_{t-\tau} + r_t, \end{cases} \tag{9.4}$$

using $\tau = 24$ to include the correlation with the load observed in the same hour of the previous day. The parameter ρ is optimized as a hyperparameter on a cross-validation basis. The model is estimated on primal space using the fixed-size version (6.23).

9.2.2 PL-AR-NARX

Consider the partially linear parameterizations PL-NARX (6.9) estimated in primal space by using (6.25), and the PL-AR-NARX formulation (6.11) estimated through (6.27). By using these structures, it is possible to observe the different performances which can be obtained by using different regressors as linear or nonlinear. Since the seminal work of partially linear models applied to electricity prices [25], it has been common practice to separate the inputs which are the past values of the load, from those inputs which are calendar and temperature effects. In the notation used here, starting from the original regression vector $\boldsymbol{z}_t$ from (8.1), the following partitions are defined to be used in the PL-AR-NARX models:

- Use the past values of the load $y_{t-i}, i = 1, \ldots, 48$ as nonlinear regressors, and the exogenous inputs $\boldsymbol{u}_t$ (calendar and temperature effects) as linear regressors. This is, $\boldsymbol{z}_{A,t} = \boldsymbol{u}_t, \boldsymbol{z}_{B,t} = [y_{t-1}; y_{t-2}, \ldots, y_{t-48}]$.
- Use the past values of the load $y_{t-i}, i = 1, \ldots, 48$ as linear regressors, and the exogenous inputs $\boldsymbol{u}_t$ (calendar and temperature effects) as nonlinear regressors: $\boldsymbol{z}_{A,t} = [y_{t-1}; y_{t-2}, \cdots, y_{t-48}], \boldsymbol{z}_{B,t} = \boldsymbol{u}_t$.

The corresponding PL-AR-NARX models are estimated from these formulation by also including a correlation structure with $\tau = 24$ as in the case of (9.4).

9.3 Methodology

A description of the methodology for model estimation and final assessment is described on this section.

9.3.1 Available data

A set of 4 different load series, each one containing 36,000 datapoints for the training set, is considered for this exercise. The linear PAR models are estimated individually for each series. Following the same procedure as in Chapter 8, a first linear regression is estimated for each series to remove any growth trend existing in the data. Each series is normalized using the sample average and standard deviation.

9.3.2 Implementation using Fixed-Size versions

The size M of the subsample from which the feature map approximation is computed can affect the performance of the final model. For this implementation we use $M = 1000$, which accounts for 4% of the available dataset. As observed on Chapter 8, the size of $M = 1000$ can produce very good results for this problem without requiring too much computational effort. Larger values for M improve the results only marginally at a higher computational cost. For every load series, the selection of support vector is made by maximization of the quadratic Renyi entropy.

The tuning of the RBF kernel hyperparameter σ, the regularization term γ and the AR coefficient ρ are performed by 10 fold cross-validation (CV) in the training sample. For a given kernel function (built with σ and ρ using the equivalent kernel formulations given on Table 7.4) and a given regularization parameter γ, the training sample is divided in 10 parts, 9 of which are used for model estimation. The performance of the estimated model is assessed in the remaining data part (used as a test set). By repeating the process over the 10 parts, the cross-validation performance of the model is the average of the 10 individual performances. The whole process is repeated for different hyperparameters.

9.3.3 Performance Assessment

The final models, linear and nonlinear, are evaluated on different test samples (not part of the training set), where their performance is measured by the Mean Squared Error (MSE). Usually models for short-term load forecasting may be used for prediction during a certain number of days, after which the models are re-estimated with the new information available. Typically a model might be re-estimated after no more than one week, in some cases in a matter of a few days. Rarely a model remains valid for more than 7 days without re-estimation. In this context, for the purposes of evaluation of the models in this study, 50 different non-overlapping weeks are used to assess the model performances. Denote by d the last day contained in the training dataset. The first test period is the week occurring immediately after the training data, going from $d+1$ until $d+7$. The second test week begins at $d+8$, finishing at $d+14$, the third period begins at $d+15$ until $d+21$, and so on, until the last test period which begins at $d+344$ and ends on $d+350$. The model performances are assessed by using MSE on each of the test periods, after which the average and standard deviation are reported for each case.

9.4 Results

In this section, the empirical results are presented. First, an individual description for linear and nonlinear models is presented, and a final comparison concludes the section. It is important to keep in mind that we are using a highly structured multi-equation linear model as a benchmark towards the single-equation structured nonlinear models. Therefore the PAR models probably belong to a different class of models than those comparable to the nonlinear variants of the NARX models used in this work. However, from a practical perspective it is a very strict benchmark for checking the quality of the (AR)-NARX models.

9.4.1 PAR Model

One practical advantage of linear models is that the coefficients have a direct interpretation, capturing the effect of each of the explanatory variables on the load behavior. The particular advantage of the PAR model structure

is that it provides a coefficient for each variable for each hour of the day. In this way, it is possible to compare the behavior of different load series by looking into the estimated values of parameters of interest. One example is the set of parameters related to temperature $CR_{h,d}$,$HR_{h,d}$, $XHR_{h,d}$; the difference between corresponding parameters gives information on temperature sensitivity.

Figure 9.1 shows the results for heating and cooling requirements ($HR_{h,d}$ and $CR_{h,d}$, respectively), where their estimated coefficients are depicted as bars. Usually, it is accepted that heating and cooling requirements increase the energy consumption; this is shown by Series 1 in Figure 9.1.

However, other series exhibit a different sensitivity. Series 2 does not seem to show temperature sensitivity for cooling. Series 3 and 4 show that the maximum effects from temperature-related variables can occur at different hours of the day.

The forecasting performance of the PAR models is very satisfactory in terms of the one-hour-ahead predictions and the 24-hour-ahead simulations. Figures 9.2 and 9.3 illustrate the performance of the PAR models for the different forecasting modes for Series 1 and Series 2 for one of the test weeks. The actual load (thin line) is compared to the one-hour-ahead predictions (thick line, top panel) and the 24-hours-ahead simulations (thick line, bottom panel).

9.4.2 AR-NARX

The autocorrelation parameter ρ is tuned by minimizing the cross-validation MSE, obtaining an optimal ρ^* close to -0.4 for Series 1, as shown in Figure (9.4). According to this result, a negative correlation on the residuals of the NARX model can be captured using the AR-NARX parameterization.

9.4.3 PL-AR-NARX

Finally, it is still possible to check the effect of imposing an autocorrelation structure to the partially linear models using AR-PL-NARX formulations. Using the partitions over the original regression vector defined above, two models are estimated, PL-AR-NARX-1 and PL-AR-NARX-2.

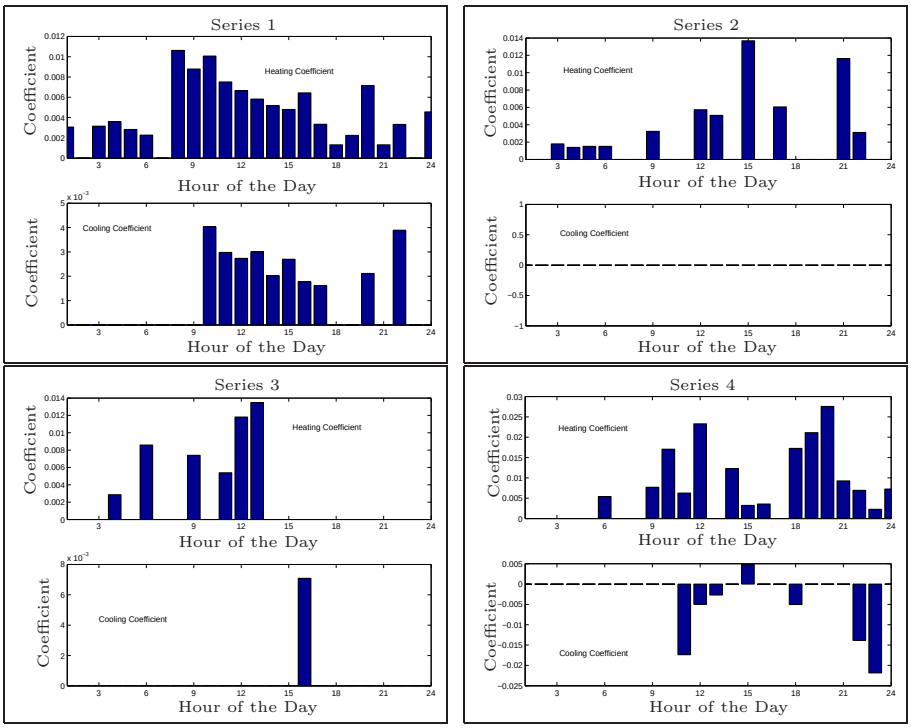

Figure 9.1: *Parameters Identification with PAR models. The estimated coefficients for Heating (top) and Cooling (Bottom) requirements show different types of sensitivities across substations. Those with zero value are not statistically significant. Maximum effects can occur at different hour of the day, and also with different sign: sometimes cooling increases the load, sometimes the need for cooling has a decreasing effect.*

In the case of the PL-AR-NARX-1 model (which uses $\boldsymbol{z}_{A,t} = \boldsymbol{u}_t, \boldsymbol{z}_{B,t} = [y_{t-1}; y_{t-2}, \ldots, y_{t-48}]$), the optimal ρ^* is found to be -0.2, while in the case of the PL-AR-NARX-2 model (which uses $\boldsymbol{z}_{A,t} = [y_{t-1}; y_{t-2}, \ldots, y_{t-48}]$, $\boldsymbol{z}_{B,t} = vecU_t$), $\rho^* = -0.5$ for Series 3, as shown in Figure 9.7.

The PL-AR-NARX models contain a subset of linear regressors. Therefore it is possible to explore the coefficients of the explanatory variables of interest. For example, PL-AR-NARX-1 model contains a linear set of coefficients for the calendar and temperature variables. Figure 9.8 shows the coefficients for calendar (monthly) effects for Series 2 (relative to the effect of December). This series has a clear winter-summer cycle, where the

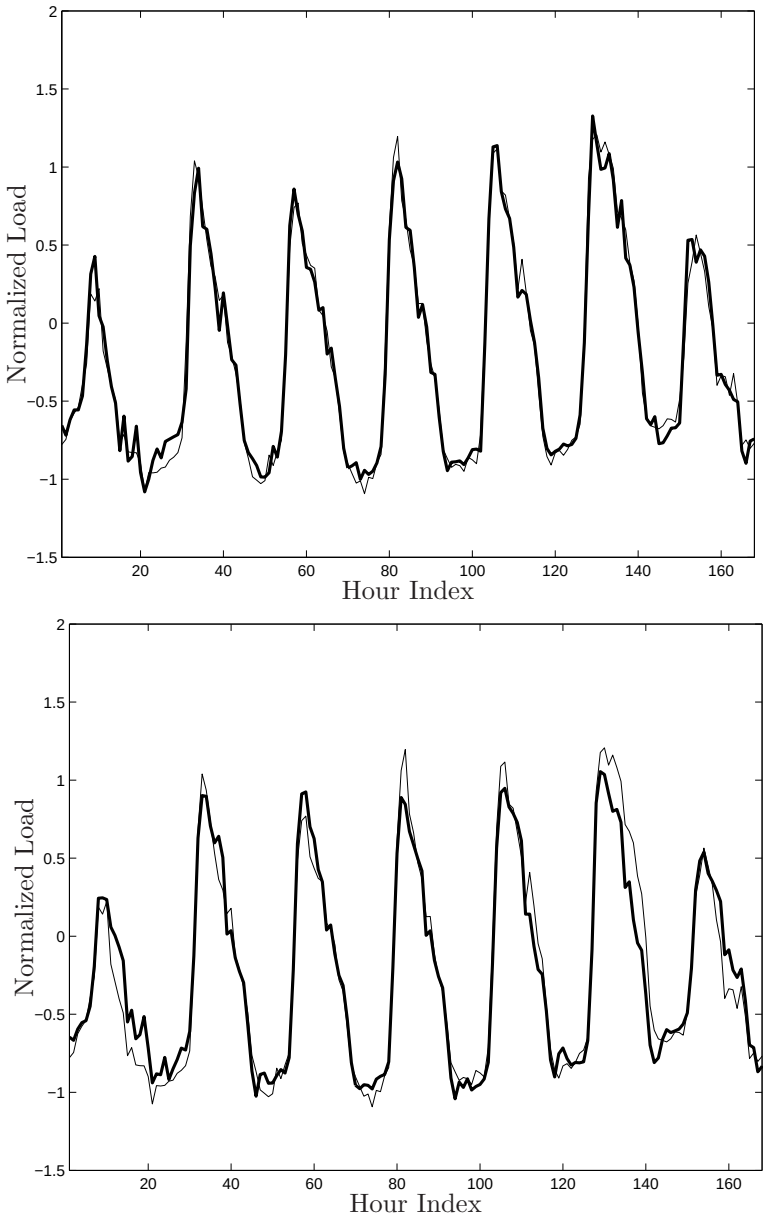

Figure 9.2: *Prediction performance of the PAR model for Series 1 for a week in the test set. The actual load (thin line) is compared to the predictions from the PAR models (thick line) for the one-hour-ahead prediction mode (top) and the 24-hours-ahead simulation mode (bottom).*

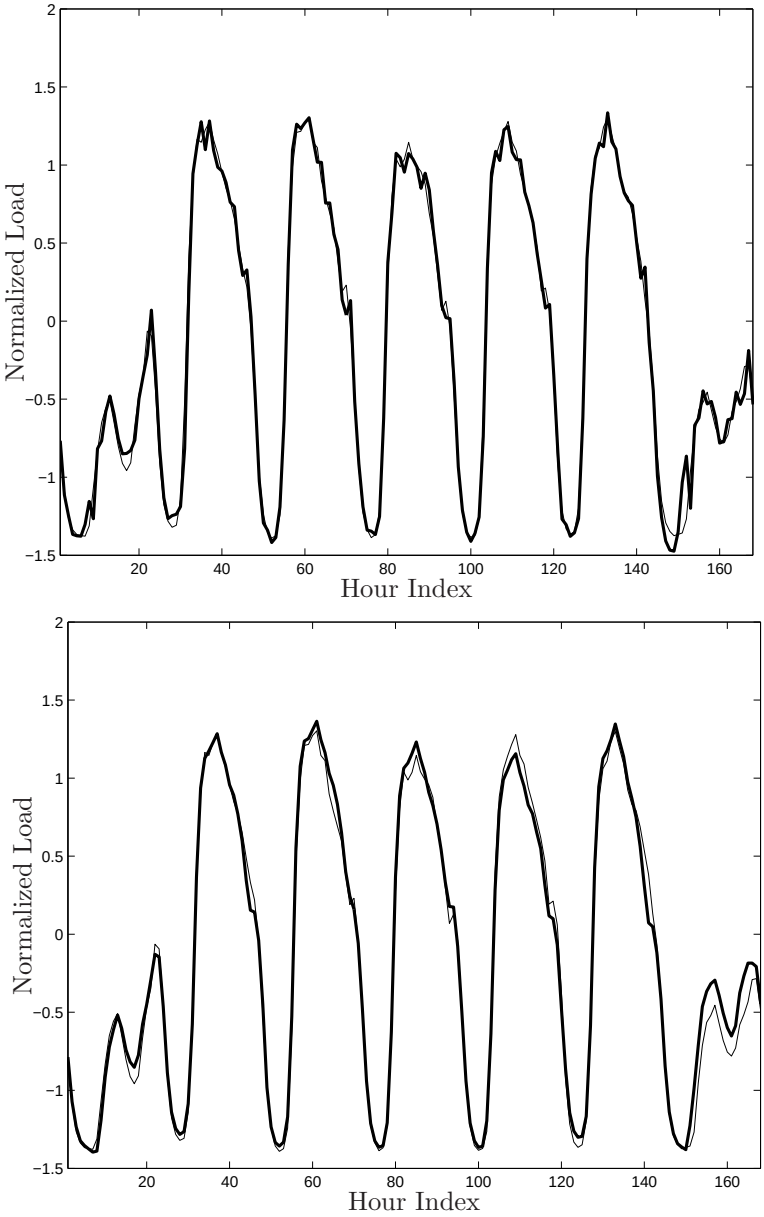

Figure 9.3: *Prediction performance of the PAR model for Series 2 for a week in the test set. The actual load (thin line) is compared to the predictions from the PAR models (thick line) for the one-hour-ahead prediction mode (top) and the 24-hours-ahead simulation mode (bottom).*

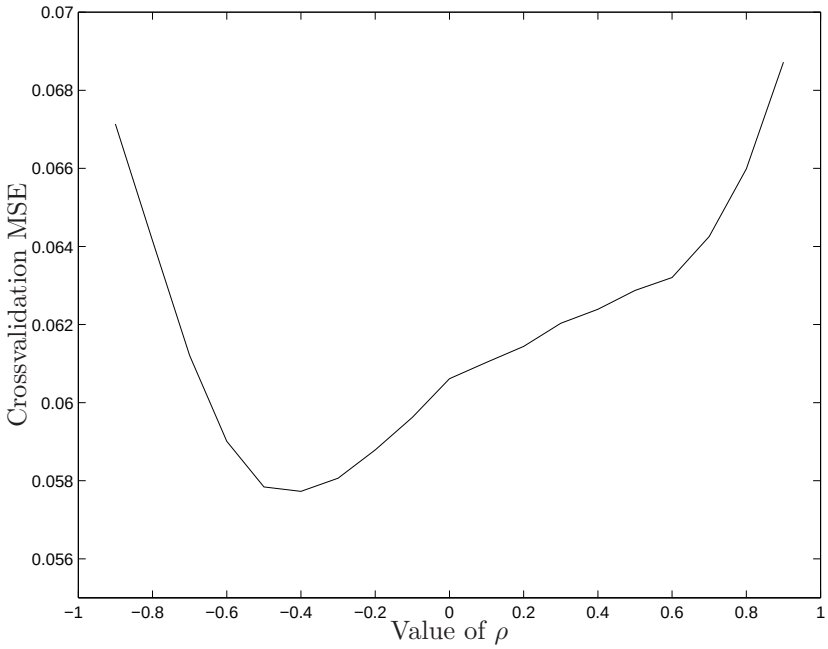

Figure 9.4: *Cross-validation performance of the AR-NARX model for Series 1 for different values of* ρ.

lower bars are located in the summer months, and only the January effect is slightly higher than that of December (reference level). Although the same exercise can be done with a purely linear model, the PL-NARX structure makes sure the parameters are estimated in a better way (Chapter 4) by taking into account the nonlinear effects of the other regressors. Figure 9.9 shows the same exercise for the case of Series 3, where the effects from January until March still show an increase in the load over the December value. Surprisingly, the effect of October is also above the reference level of December. It is important to remember that these effects have been cleaned from the weather variations (which, in turn, obtain their own set of parameters). Therefore, these monthly effects are related to non-weather seasonal patterns, like e.g. more daylight in summer and holiday periods.

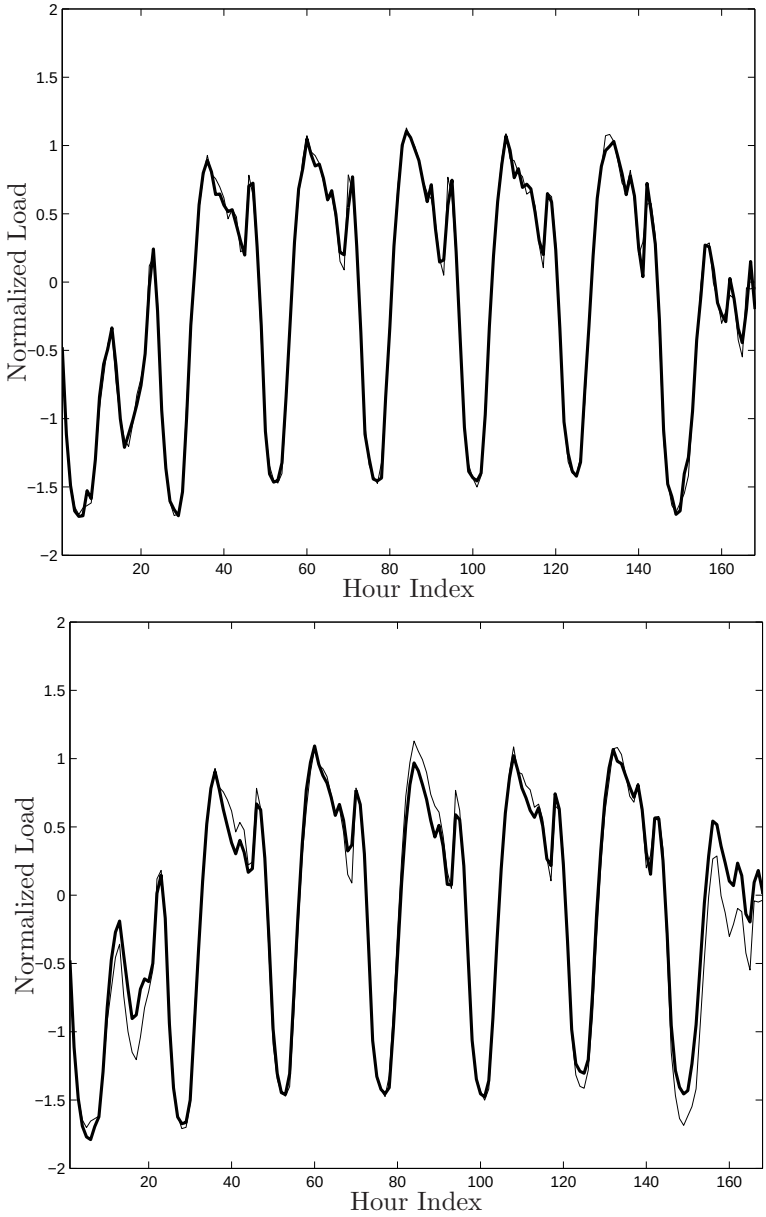

Figure 9.5: *Prediction performance of the AR-NARX for Series 3 for a week in the test set. The actual load (thin line) is compared to the predictions from the AR-NARX (thick line) for the one-hour-ahead prediction mode (top) and the 24-hours-ahead simulation mode (bottom).*

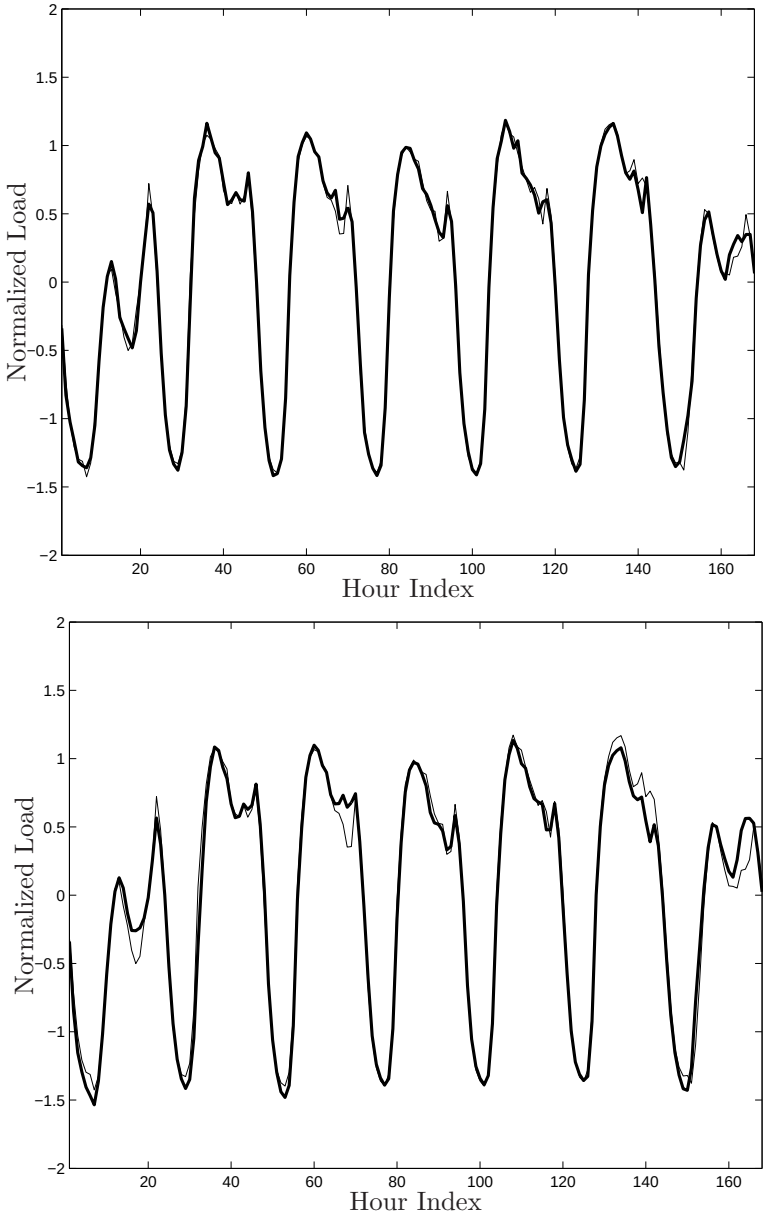

Figure 9.6: *Prediction performance of the AR-NARX model for Series 4 for a week in the test set. The actual load (thin line) is compared to the predictions from the AR-NARX models (thick line) for the one-hour-ahead prediction mode (top) and the 24-hours-ahead simulation mode (bottom).*

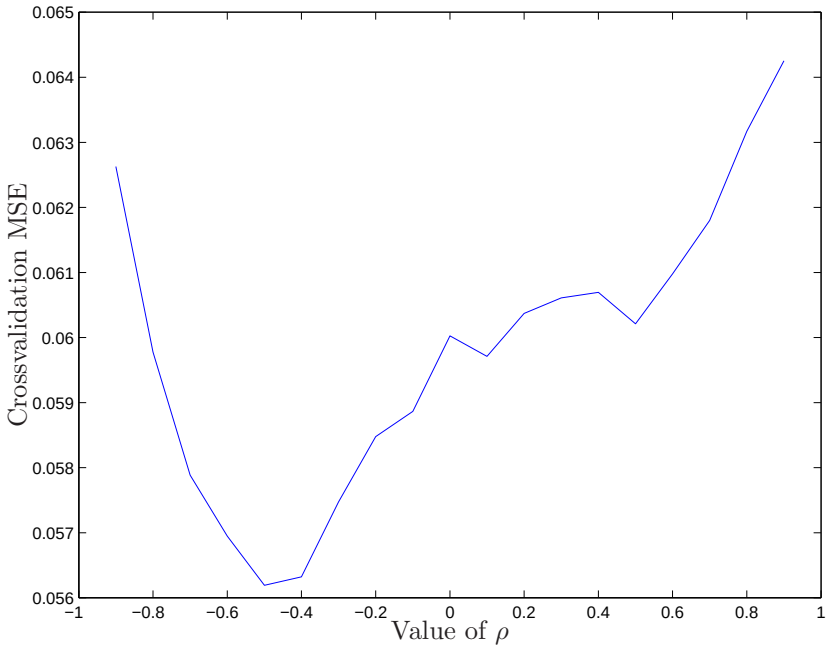

Figure 9.7: *Cross-validation performance on Series 3 of the PL-NARX-2 model for different values of* ρ.

9.4.4 Forecasting Comparison

The forecasting performance of the different models is reported in Table 9.1 for the case of one-hour-ahead predictions. The performance of the PAR models is not as good as those from the nonlinear models. The comparison across nonlinear models shows consistent results. In general, it is possible to conclude that the addition of the autocorrelated noise to a model structure improves the results over the case where autocorrelation is not included. In some cases this improvement is small or even non-existing, as in the case of the change between the NARX and AR-NARX models for Series 3. However, for the partially linear structures, a substantial improvement is observable, particularly for the PL-NARX-2 models. Using a partially linear structure gives the user the freedom to experiment with different set of regressors, particularly using a linear term for certain parameters of interest. Although the partially linear models do not achieve the excellent performances of the NARX models for these load series, clearly they improve substantially

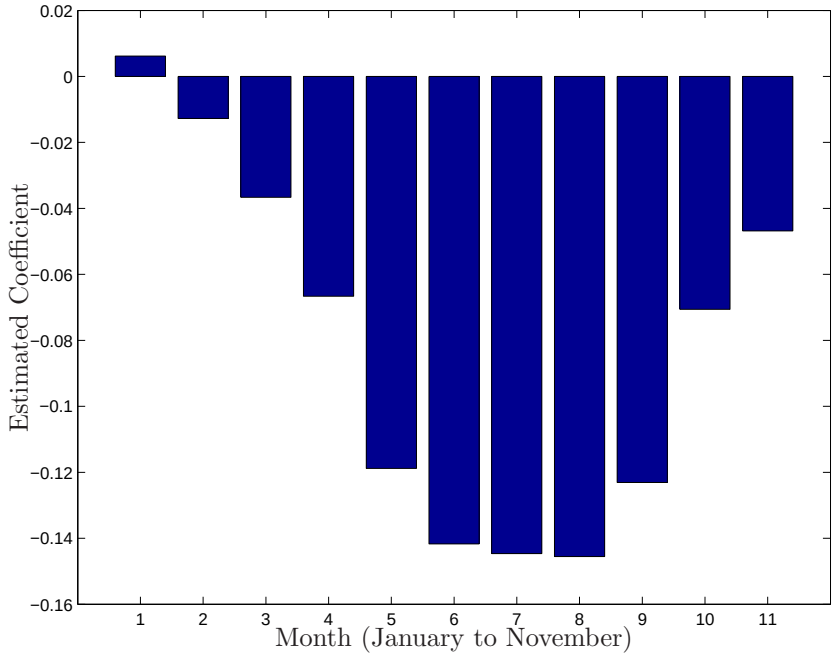

Figure 9.8: *Monthly effects captured by the PL-AR-NARX-1 model for Series 1. Each bar shows the contribution due to "being in January" (1st column), "being in February" (2nd column), until "being in November" (11th column). Each bar is relative to the month of December (level 0). A month with a negative value, indicates that its associated effect contributes to a lower value of the load than the one observed in December.*

over the linear models, even for the highly structured linear models. When the goal of modelling is not only prediction accuracy, but also descriptive information about load series, the partially linear models (with AR noise) can give the best of both worlds. The results are depicted on Figures 9.10, 9.11 and 9.12.

For the 24-hour-ahead simulation mode, the comparison is made between unstructured and structured models. Table 9.2 shows the big improvement in the linear models when moving from an unstructured ARX (as the one used in Chapter 8) towards a PAR model. For Series 3 and 4, the PAR model produces even better results than the fully black-box NARX model. However, the NARX models also improve when including the structure in the residuals. The inclusion of the autocorrelated residuals as defined in

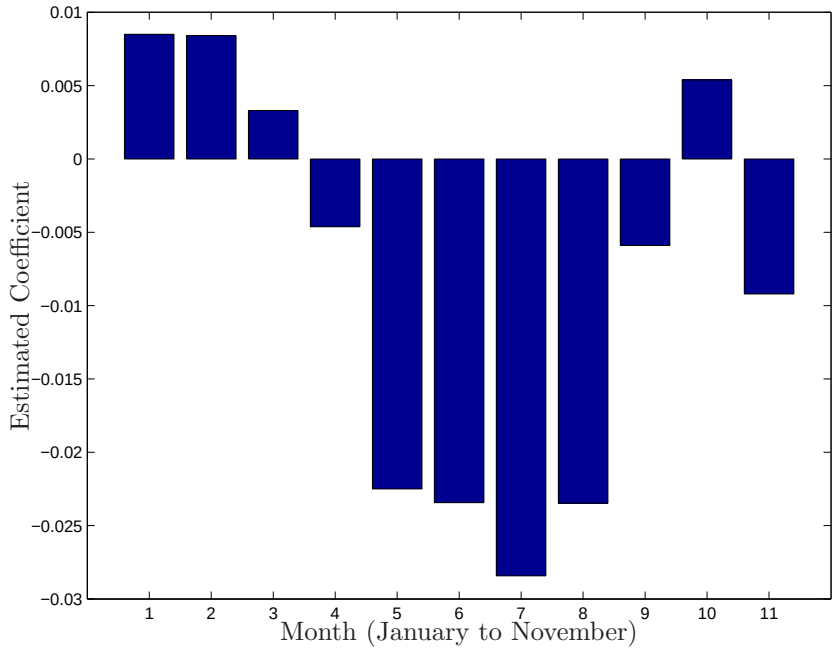

Figure 9.9: *Monthly effects captured by the PL-AR-NARX-1 model for Series 3. Each bar shows the contribution due to "being in January" (1st column), "being in February" (2nd column), until "being in November" (11th column). Each bar is relative to the month of December (level 0). A month with a negative value, indicates that its associated effect contributes to a lower value of the load than the one observed in December.*

(9.4) produces an improvement in such way that the final nonlinear model outperforms the PAR model as well. This set of results is also depicted on Figures 9.13 and 9.14.

9.5 Conclusion

For the problem of short-term load forecasting, it has been shown that the use of structured models can improve over the case of the ARX and NARX models of Chapter 8. In the linear case, the Periodic Autoregressive (PAR) models provide a highly-structured linear parameterization for this problem, substantially improving the performance over the one of the linear ARX

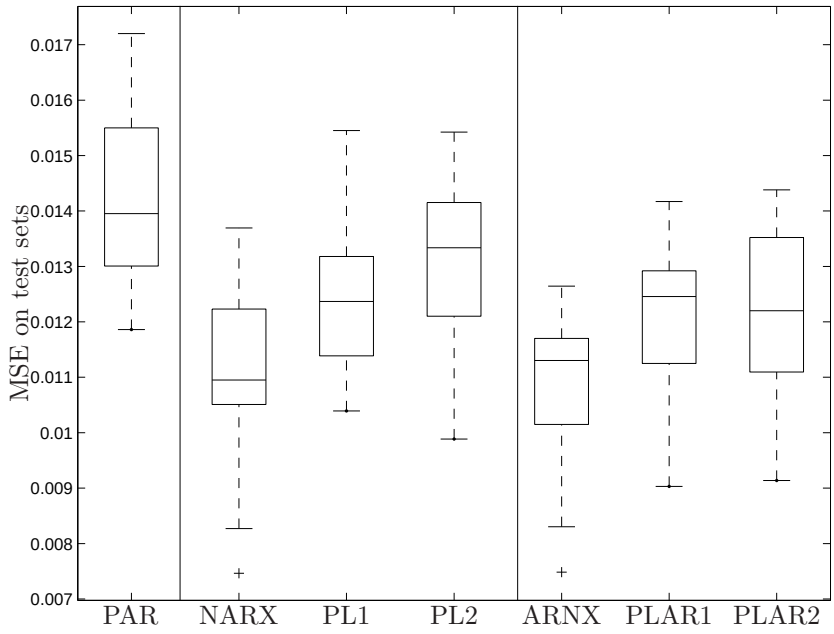

Figure 9.10: *One-hour-ahead forecasting performance over 50 different test weeks for all the models, for Series 1. The PAR model provides the worst performance; the unstructured NARX model is already much better. The inclusion of the autocorrelated residuals improves the model performances, particularly for the PL-NARX-2 model.*

		Without AR noise			With AR noise		
	PAR	NARX	PL-1	PL-2	NARX	PL-1	PL-2
1	0.0142	0.0110	0.0124	0.0130	0.0107	0.0122	0.0120
	(0.0016)	(0.0018)	(0.0014)	(0.0017)	(0.0016)	(0.0015)	(0.0017)
2	0.0180	0.0081	0.0107	0.0120	0.0074	0.0086	0.0100
	(0.0035)	(0.0018)	(0.0005)	(0.0018)	(0.0012)	(0.0008)	(0.0021)
3	0.0115	0.0057	0.0060	0.0090	0.0057	0.0060	0.0081
	(0.0016)	(0.0016)	(0.0012)	(0.0013)	(0.0015)	(0.0014)	(0.0013)
4	0.0140	0.0073	0.0081	0.0106	0.0071	0.0081	0.0101
	(0.0040)	(0.0017)	(0.0022)	(0.0026)	(0.0017)	(0.0021)	(0.0028)

Table 9.1: *Results for the 4 load series and the different model structures for the case of one-hour-ahead forecasting mode over 50 different test sets. The average MSE is reported, and the standard deviation is given in brackets.*

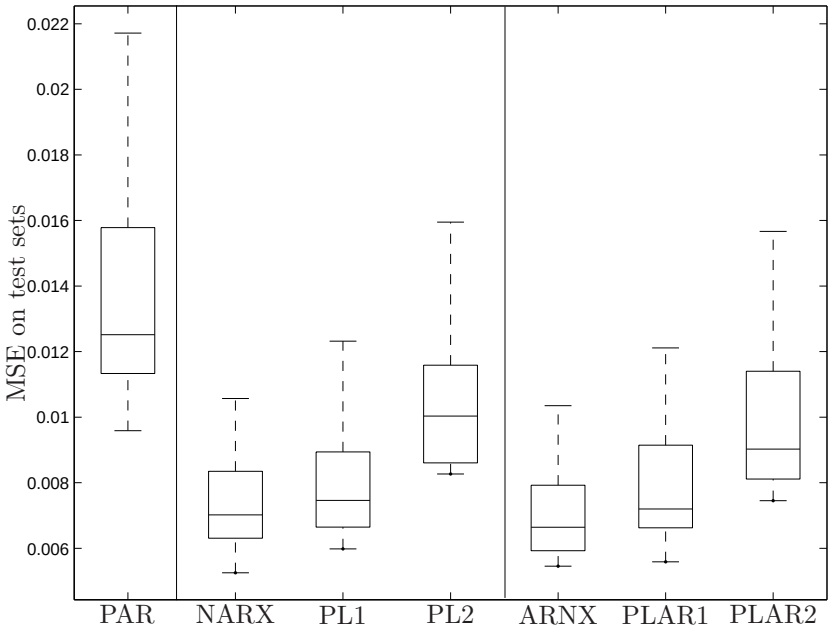

Figure 9.11: *One-hour-ahead forecasting performance over 50 different test weeks for all the models, for Series 4. The PAR model provides the worst performance; the unstructured NARX model is already much better. The inclusion of the autocorrelated residuals improves the model performances, particularly for the PL-NARX-2 model.*

models. The PAR models require the estimation of a set of 24 equations, one for each hour of the day, in such a way that each input variable obtains an estimated coefficient varying across 24 hours. As this is a linear model, it provides interpretable results for each of the variables of interest.

The nonlinear models, on the other hand, have been estimated including an autocorrelation with the residuals lagged 24 hours. The large sample size makes it necessary to estimate the models in primal space, for which the support vectors are selected by maximizing the quadratic Renyi entropy with a subsample of 1000 datapoints. The parameter of the autocorrelated residuals is tuned as a hyperparameter under a cross-validation procedure. This increases the computational cost of the models, although everything can be done in such a way that no user interaction is required.

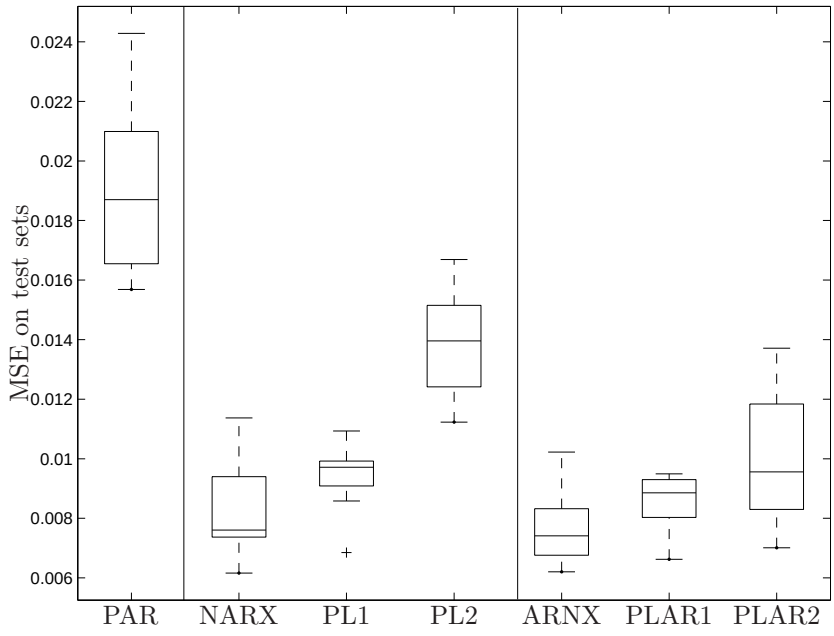

Figure 9.12: *One-hour-ahead forecasting performance over 50 different test weeks for all the models, for Series 2. The PAR model provides the worst performance; the unstructured NARX model is already much better. The inclusion of the autocorrelated residuals improves the model performances, particularly for the PL-NARX-2 model.*

The models are compared on a one-hour-ahead forecasting mode, where the nonlinear models show better results than the PAR model. In addition, adding the extra autocorrelation to the residuals on each of the corresponding nonlinear structures (NARX, PL-NARX-1 and PL-NARX-2), does not deteriorate the results. On the contrary, particularly for the partially linear structures, the results show substantial improvements, bringing them closer to the fully black-box NARX models. This result is of major practical importance, as it shows that the partially linear structures can obtain forecasting performances similar to a fully nonlinear model, yet keeping some regressors in linear form in such a way that there is a set of interpretable coefficients for the variables of interest.

Finally, a comparison is made in a 24-hours-ahead simulation mode, for the case of structured versus unstructured models. The improvement from

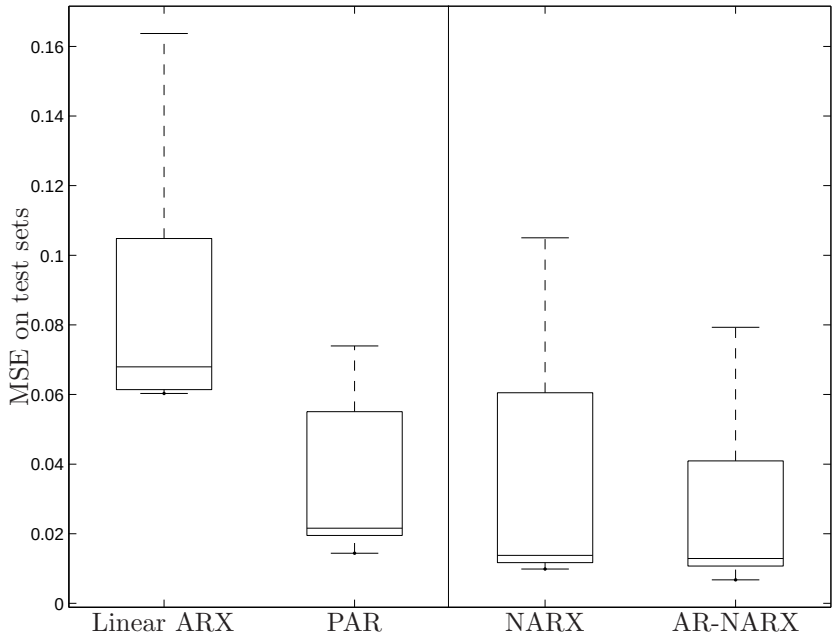

Figure 9.13: *24 hours ahead simulation performance over 50 different test weeks. The performance of the (unstructured) linear ARX model improves when changing to a structured PAR formulation. The NARX model also improves its performance when including the structured autocorrelated part. Results shown for Series 2.*

an unstructured linear ARX to a highly structured PAR model is clearly observable. However, the NARX model improves towards a AR-NARX formulation, consistently obtaining the best results over the different test sets and load series. From a practical perspective, the use of PAR models may require more supervision or direct user interaction. The nonlinear models implemented for this chapter can take more computing time, but almost no interaction is required. The framework of nonlinear models using different structures can lead to further improvements not only on load forecasting, but also as support for long term planning and scenario analysis.

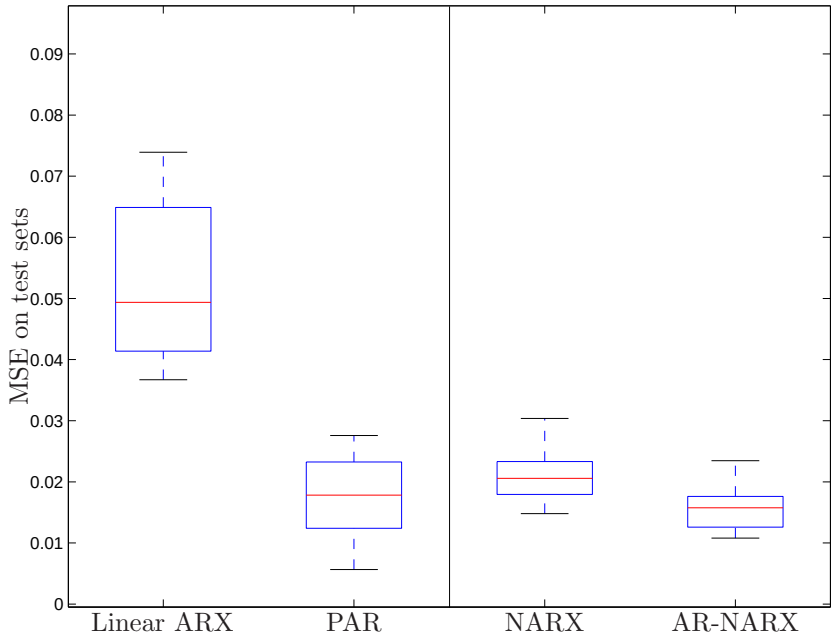

Figure 9.14: *24 hours ahead simulation performance over 50 different test weeks. The performance of the (unstructured) linear ARX model improves when changing to a structured PAR formulation. The NARX model also improves its performance when including the structured autocorrelated part. Results shown for Series 4.*

Series	Linear Models		Nonlinear Models	
	ARX	PAR	NARX	AR-NARX
Series 1	0.0628	0.0447	0.0401	0.0400
	(0.0197)	(0.0184)	(0.0152)	(0.0171)
Series 2	0.0843	0.0447	0.0338	0.0301
	(0.0337)	(0.0238)	(0.0248)	(0.0237)
Series 3	0.0874	0.0390	0.0468	0.0309
	(0.0365)	(0.0307)	(0.0386)	(0.0320)
Series 4	0.0572	0.0241	0.0267	0.0221
	(0.0243)	(0.0256)	(0.0237)	(0.0228)

Table 9.2: *Summary of Results for the 4 load series and the selected model structures for the case of 24-hour-ahead simulation mode over 50 different test sets. The average MSE is reported, and the standard deviation is given in brackets. This table shows the improvement when using structured models.*

Chapter 10

General Conclusions

10.1 Concluding Remarks

The research described on this thesis covers a series of topics related to applied nonlinear modeling of time series. First, nonlinear estimation techniques are presented, with a modular design for different cases of nonlinear regression structures. Second, these techniques are incorporated in the framework of nonlinear system identification, leading to practical implementations for large scale problems, with very good results in terms of forecasting ability. Third, the nonlinear system identification techniques are further used in the context of the real-life problem of Short-Term load forecasting.

The first part of the thesis presents the results in the context of regression estimation. Taking as starting point the Least-Squares Support Vector Machines formulation for nonlinear regression, this thesis extends the framework by considering the following cases:

- Imposing symmetry to the nonlinear function estimated with LS-SVM (Chapter 3)
- Imposing an additional parametric term for a new set of regressors (Chapter 4), and
- Incorporating autocorrelation in the noise process of the nonlinear regression (Chapter 5).

All these extensions over the standard LS-SVM regression have been formulated and discussed, including their links with related techniques from statistics and/or econometrics. For each extension, the goal has been to include the additional structures, which may be interpreted as prior knowledge, in the form of equality constraints such that the least-squares optimization problem remains convex, and Mercer's theorem can be applied to move from primal to dual space. An important conclusion is that the new information contained in the additional constraints becomes embedded at the kernel level. This equivalent kernel contains the information imposed to the training datapoints, and it can be used directly to evaluate the models for new datapoints without having to include the restrictions again. The equivalent kernel makes an important contribution in terms of modularity of the model formulation, in the sense that different types of prior knowledge can be tested in practice simply by changing the kernel function being used. In addition, the large scales versions of the different extensions of the LS-SVM can be formulated in primal space by using the Nyström method in the same way as in the standard LS-SVM, as the approximation is built from any kernel matrix. Therefore, this kernel matrix can be built with any of the obtained equivalent kernels, embedding the prior knowledge into the approximation in primal space.

The second part of the thesis considers the nonlinear system identification framework. By considering each one of the developed LS-SVM extensions as building blocks, a modular approach for the case of nonlinear system identification is proposed (Chapter 6). It has been shown that this framework can be used for the estimation of NARX and AR-NARX model structures, with different possible parameterizations. This framework also provides a practical approach for moving gradually from a fully nonlinear black-box model towards a linear model, being very important in applied work. In addition, there is the practical advantage of formulating the model in dual space and computing the estimation in primal space for very large sample sizes, by using the equivalent kernel representation. Practical examples for chaotic time series and the SilverBox dataset (Chapter 7) show the merits of this methodology in applied work, where large scale problems are successfully tackled using nonlinear regressions formulated in dual space and estimated in primal space.

The third part of the thesis shows a real-life industrial application of the methods developed in the previous two parts. The nonlinear system identification methods are tested for the case of the short-term electric load

forecasting problem. Large time series are available for this task, leading to the estimation of the models in primal space. A first comparison between a linear ARX and a nonlinear black-box NARX model, both formulated with the same set of inputs, shows that the nonlinear model can capture the behavior of the load series and generate more accurate forecasts than the linear one, on one-hour-ahead and on 24-hours ahead basis, which was verified empirically for 10 different load series. A second step in this comparison has involved the formulation of more structured models, both linear and nonlinear. A linear Periodic Autoregressive (PAR) formulation, being a highly structured set of 24 hourly equations, is compared to different parameterizations of AR-NARX models, where the structure is embedded in the 24-hours correlation of the residuals. The results show the benefits of including structure into the models. The improvement of the PAR linear model is seen in the performance of the 24-hours-ahead simulations. The nonlinear models can improve over the highly structured multi-equations linear PAR models when the 24 hours correlation is taken into account. This shows that structured single-equation nonlinear models can produce more accurate forecasts on a one-hour-ahead basis, and they also can produce more accurate 24 hours-ahead simulations than their linear counterparts. From the practical point of view, it requires more practical expertise to define and estimate a PAR model than any of the AR-NARX models based on the solution of a linear system and where the required hyperparameters are tuned by cross-validation.

The work described in this thesis starts from a theoretical perspective and gradually descends towards practical examples and real-life industrial applications. The modular approach proposed in this thesis is also reflected in its chapter structure, where previous chapters are used to build the subsequent chapters. It has been shown that such as approach can be quite successful in the definition, estimation and final forecasting performance of nonlinear time series models for real-life problems.

10.2 Future Research

The research presented in this work can be further extended in several directions.

- One research topic is related to shifting from “Imposing” to “Detect-

ing". In the proposed framework, structured elements (symmetry, correlated noise) can be imposed to the NARX models when the user has prior knowledge about them. However, it is yet unclear how to use this framework for detection of symmetry, or correlated residuals for a given dataset.

- Another direction for future research is related to the hyperparameters selection procedure. The current selection of hyperparameters is performed following a training-validation scheme. However, this procedure is time consuming. It works well for a limited number of hyperparameters to be tuned, but it becomes a limiting factor when the number of hyperparameters grows. In such case, this approach is often too time consuming in practice. In the framework presented in this thesis, the use of correlated noise has been developed theoretically for a general AR(q) process containing q parameters to be identified, but in practice this is only implemented as AR(1) because the parameters are tuned in a cross-validation basis. As the correlation parameters in the noise models get into the kernel function, the kernel matrix has to be built before the other parameters of the model are estimated. If, on the other hand, the correlation parameters are considered to be unknowns at the same level as the lagrange multipliers and the bias term, then the problem becomes nonconvex. In this sense, future research involves the development of an efficient optimization method for this possibly nonconvex problem in order to be able to estimate models beyond the AR(1).

- Another line for future research involves the extensions towards other model structures. Not only NARX or AR-NARX models, but also nonlinear output-error models (NOE), nonlinear Box-Jenkins (NBJ), nonlinear ARMAX models, etc. Currently only past values of inputs and outputs are used as regressors in the nonlinear models presented in this thesis. The other model structures require the use of past predictions and/or past residuals in the model formulation, which leads to recurrent models and/or nonconvex optimization formulations.

- It is also important to incorporate the quantification of the error for the predictions. Existing techniques for error bars, based on probabilistic assumptions and/or dedicated bootstrap methods, are derived only for the standard LS-SVM regression. It is interesting to study the effect of imposing the extra constraints into the error bars quantification. Imposing more structure into the model may translate in reducing the

prediction errors.

- Finally, the implementations in the case of load forecasting can be improved for more specific applications. Other real-life problems show similar properties of seasonal behavior and large datasets available, e.g. gas consumption, utilities, internet traffic, logistics and distribution, and others. In these contexts, knowledge extraction, identification of local weather sensitivities, applications for decision support, hedging and pricing mechanisms, risk management on energy prices, are all applications which can benefit from the framework presented on this thesis.

Appendix A

Clustering Load Series using PAR representations

The multi-equations structure of the Periodic Autoregressive (PAR) models used in Chapter 9 can be used to compute a Typical Daily Profile, leading to a representation which can be used for clustering. This appendix discusses the implementation of a clustering exercise using the PAR models.

A.1 Clustering of Customer Profiles

Within the electricity sector the need for strategic information has become extremely important. Not only accurate forecasts are needed for the short-term operations and mid-term scheduling, but also network managers need to have insight in the type of customers they have to supply as a support for long-term planning. The unbundling between generation, transmission, distribution and supply induced by the market liberalization has led to network managers being partially blind beyond a certain substation level with respect to the final customers. In these cases, it is required to use indirect techniques to assess the type of demand they have to face [58, 75] in order to support their long-term investment planning. In this context, categories of residential, business and industrial customers have been documented for some locations [15, 66] and usually they are recognized by their "typical" load pattern over a day.

The two problems outlined above, forecasting and profiling, usually have been tackled independently. However, from a manager point of view, the boundary between both problems is irrelevant, and eventually unnecessary. Given a set of measurements, it is possible to produce accurate short-term forecasts and at the same time generate a tool for extracting information on the overall demand structure. In this chapter, it is shown how to identify and remove the influence of temperature fluctuations and how to use the forecasting model to identify the type of customer being modelled. This methodology is demonstrated on a set of 245 time series provided by the Belgian National Grid Operator ELIA, details of which are described below. The methodology is oriented towards handling the problems of short-term forecasting and profile segmentation in a unified framework based on the PAR models described on Chapter 9. By exploiting the structure of the PAR models, a smooth transition from a forecasting towards a clustering problem is achieved.

A.2 Typical Daily Profiles

The definition of a Typical Daily Profile for each substation from the parameters of the system (9.3) is described in this section.

A.2.1 Equivalent Vectorial Notation and Convergence

By defining a vector $\boldsymbol{y}_d = [y_{1,d} \quad y_{2,d} \quad y_{3,d} \quad \cdots \quad y_{23,d} \quad y_{24,d}]^T \in \mathbb{R}^{24}$, containing the load information for the 24 hours of day d, it is possible to write (9.3) as

$$\Phi_0 \boldsymbol{y}_d = \boldsymbol{c} + \Phi_1 \boldsymbol{y}_{d-1} + \Phi_2 \boldsymbol{y}_{d-2} + \Phi_3 \boldsymbol{X}_d + \boldsymbol{\varepsilon}_d \tag{A.1}$$

with Φ_0, Φ_1, Φ_2 and Φ_3 containing the coefficients θ of the system (9.3) [42]. The matrix $\boldsymbol{X}_d$ contains all exogenous variables for temperature and calendar information. The system is now written in a Vector Auto-Regression (VAR) form with 2 lagged values for $\boldsymbol{y}_d$ [51], and it is easily seen that the original PAR(48) is equivalent to a VAR(2) formulation. Once the system (9.3) has been estimated, all coefficients of the matrices Φ_0, Φ_1, Φ_2 and Φ_3 are known. The next-day forecasts can be simply written as

$$E_d[\boldsymbol{y}_{d+1}] = \hat{\boldsymbol{y}}_{d+1} = \Phi_0^{-1}\{\boldsymbol{c} + \Phi_1 \boldsymbol{y}_d + \Phi_2 \boldsymbol{y}_{d-1} + \Phi_3 \boldsymbol{X}_{d+1}\}$$

where E_d is the expectation taken at time d. The matrix Φ_0 is always invertible as it is a lower triangular matrix with ones in the main diagonal. Applying this formulation iteratively for n days, a multi-step ahead prediction can be obtained. The above equation requires the knowledge of the values of $\boldsymbol{X}_{d+1}$ on day d. At least the calendar information is always available for the future, and for the temperature information one should rely on the best available weather-temperature forecasts. In any case, the information contained in the variables $\boldsymbol{X}_d$ are exogenous (to the load) as they are capturing seasonal effects to the load itself. Thus, a very interesting question is to check what happens to the load when these variables are defined to be zero, or in other words, when all seasonal effects are captured and removed from the load model[1].

Setting $\boldsymbol{X}_d = 0$, the system becomes

$$\Phi_0 \boldsymbol{y}_d = \boldsymbol{c} + \Phi_1 \boldsymbol{y}_{d-1} + \Phi_2 \boldsymbol{y}_{d-2} + \boldsymbol{\varepsilon}_d. \tag{A.2}$$

If this equation is used in iterative-forecasting mode for n periods ahead, it converges, under stability conditions, to a unique value $\boldsymbol{y}^*$, which can be computed as

$$\boldsymbol{y}^* = \{\Phi_0 - \Phi_1 - \Phi_2\}^{-1} \boldsymbol{c}. \tag{A.3}$$

Convergence is achieved if and only if the VAR system (A.2) is stationary, i.e., if the equation

$$|\Phi_0 - \Phi_1 z - \Phi_2 z^2| = 0 \tag{A.4}$$

has all its roots z_i outside the unit circle [51].

A.2.2 Typical Daily Profile Definition

The convergence condition (A.4) is verified for each of the 245 substations. This is not surprising, since an autoregressive model defined with a "correct" order leads to a stationary formulation, otherwise it can not be used as a stable forecasting model. A model that does not satisfy the convergence condition should be allowed to include extra lag terms, in order to write a VAR with a higher order until it achieves stationarity. In this dataset, every load series has its own convergence vector $\boldsymbol{y}^*$, computed from the estimated

[1] Removing temperature and seasonal effects is a usual task in long-term grid management, to compute year-to-year growth trends, to identify how much of the yearly peak was due to weather, scenario analysis, etc.

model coefficients contained in Φ_0, Φ_1, and Φ_2. As each vector $\boldsymbol{y}_d$ contains daily information of the load, the $\boldsymbol{y}^*$ convergence vector, computed after all seasonal effects have been removed, can be interpreted in terms of daily load information: it contains information about the typical daily profile of the load in absence of seasonal and temperature information. Therefore, we define the Typical Daily Profile (TDP) as follows.

Definition: The Typical Daily Profile (TDP) $\boldsymbol{y}^*$ of a sample load series $\boldsymbol{y}_d \in \mathbb{R}^{24}, d = 1, \cdots, N_d$ is the convergence vector of a VAR(q) system obtained from an equivalent PAR(p) after extracting all exogenous information.

The definition requires the obtained VAR(q) system to be stationary, a condition attainable in the process of defining the order of the PAR(p) process. It is also an empirical definition, as it is based on a statistically sound procedure which is the estimation of a set of autoregressions. The TDP can be used as a representation of the corresponding substation for which a PAR(p) model was initially computed. The main advantage of the TDP is that it provides a representation, in terms of a daily load profile, which is independent of the seasonal and calendar variations present in the load. In other words, the difference between the TDP and the actual observed daily load profile for a given day is attributable only to the seasonal and calendar effects for that particular day.

A.2.3 Typical Daily Profiles in the current sample

The dataset contains information on 245 substations. Each substation can be estimated using the PAR(48) model template, and its TDP can be computed, leading to a set of 245 TDPs. To have an assessment of the difference between each TDP and its underlying load history, Figure A.1 shows 8 substations for which the TDP (thick line) is compared with actual daily load profiles observed over 500 days randomly selected from the dataset. For each substation, the TDP captures the behavior of the load that is not attributable to seasonal and calendar variations. It is also clear that the daily behavior of these substations are not the same, with peaks located at different hours of the day. Using TDPs is thus a simple and powerful procedure for comparing the profiles of substations. Once these profiles have been identified, a natural question to ask is how many different types of profiles there are in the sample. If it is assumed that a different profile means a different type of underlying customer (or group of customers), the question translates into customer segmentation, or customer clustering.

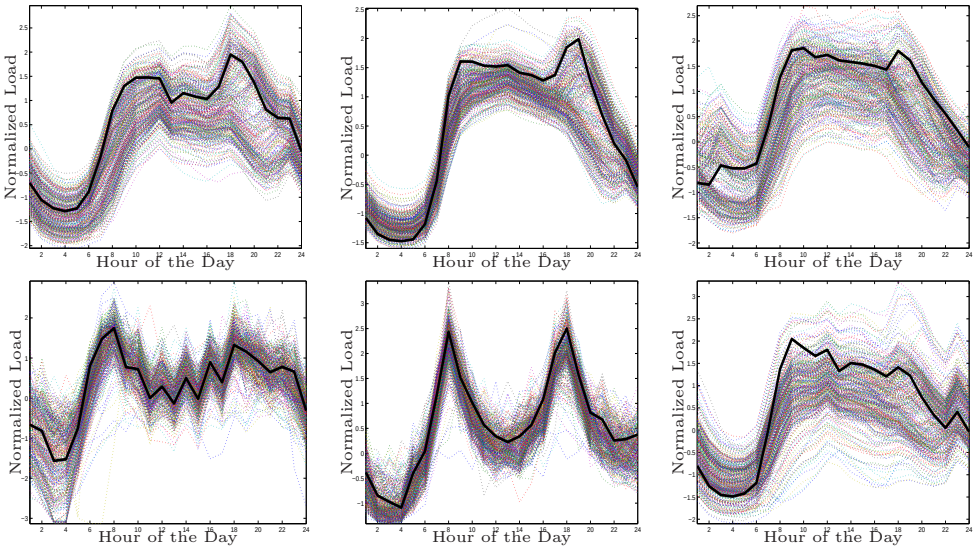

Figure A.1: *TDPs for selected 6 substations. Each panel shows the TDP of a substation (thick line) in relation to the corresponding daily loads (dotted lines) observed over 500 random days.*

A.3 Clustering using Typical Daily Profiles

This sections describes the implementation of a clustering exercise. Having identified the 245 Typical Daily Profiles (TDP), they are used as representations for the original load series. Unsupervised clustering aims at identifying different groups or patterns in a data sample, doing so without external information from the user. In this setting, the aim is to know how many different types of load profiles have to be considered, having no a priori information about the particular composition of the demand beyond each substation level.

A.3.1 Implementation

Clustering is a wide concept within statistics and machine learning [9,12,65]. In plain terms, the goal of a clustering algorithm is to identify groups of "similar" data within the dataset, without using external information, and assign each datapoint into (at least) one of the groups, or clusters. In this

application K-means is used [119], a classic clustering technique available as an standard function in many mathematical software packages. As a preprocessing step, Principal Components Analysis [68] is applied to the data in order to reduce the dimensionality of the problem. It is found that by keeping 9 principal components it is possible to recover 99% of the original set of TDPs. The application of K-means requires the user to give the number of desired clusters N_C as input parameter to the algorithm. For this case, N_C is tested from 2 to 15, which is a reasonable range, as empirical research has identified a similar number of different profiles [66,75]. In order to choose the "best" clustering, performance or validity indices are typically used [55]. In this paper the so-called Davies-Bouldin (DB) validity index [21], which is a function of the ratio of the sum of within-cluster scatter to between-cluster separation, is applied. For clusters denoted $Q_i, i = 1, \ldots N_C$, the DB index is

$$DB = \frac{1}{N_C} \sum_{j=1}^{N_C} \max_{l \neq j} \frac{S(Q_j) + S(Q_l)}{d(Q_j, Q_l)} \tag{A.5}$$

where $S(Q_k)$ is the (average) distance within cluster Q_k and $d(Q_j, Q_l)$ is the distance between clusters Q_j and Q_l. The "optimal" number of clusters N_C is the one for which the DB validity index shows a minimum value.

A.3.2 Clustering Results

A local minimum for the Davies-Bouldin validity index is found at $N_C = 8$ clusters (Figure A.2). The 8 different clusters are represented on Figure

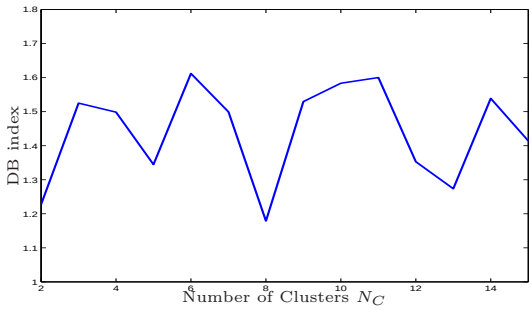

Figure A.2: Davies-Bouldin Validity Index. The local minimum at $N_C = 8$ shows that a partition containing 8 clusters can be selected.

A.3. According to interpretations by industry experts, the sample contains an important quantity of profiles with "residential" behavior, particularly clusters 1 and 5. Clusters 4 and 7 can be related to "commercial" or "business" activities. Cluster 1 captures a profile with equal peaks in the morning and evening, and a low demand in between. Clusters 3, 6 and 8 capture different variants of substation with very low demand during daytime, as e.g. street lightning or other industrial activities for which electrical energy is used at night. Possibly a more detailed characterization of the profiles based on the clustering exercise can be achieved by applying more complex techniques, or by defining an ad-hoc clustering technique for load profiles, to take into account e.g. the unbalanced presence of different profiles in the sample. Although the present exercise can be a start for industry managers to draw conclusions on the current sample, it is certainly an interesting research topic for further development.

A.4 Conclusion

The general problems of short-term load forecasting and profile identification can be addressed within a unified framework by using the proposed methodology based on the use of Periodic Autoregressive (PAR) models. By exploiting the stationarity properties of the PAR model, it is possible to compute a convergence vector that can be interpreted as a Typical Daily Profile. This convergence vector is computed from the estimated coefficients of the PAR model. This methodology is successfully applied within a sample of 245 substations. After individual PAR models are estimated, their convergence vectors are computed and the original sample can now be represented by 245 Typical Daily Profiles. This set of 245 Typical Daily Profiles can be used for clustering, in order to quantify how many different groups or classes of profiles can be identified within the sample. Using a classic clustering technique, it is possible to identify 8 different clusters, capturing the different types of profiles within the sample. This information can be used for further specific refinement of the forecasting models, such as building ad-hoc models for each specific cluster. It can be a starting point for the application of more complex clustering techniques.

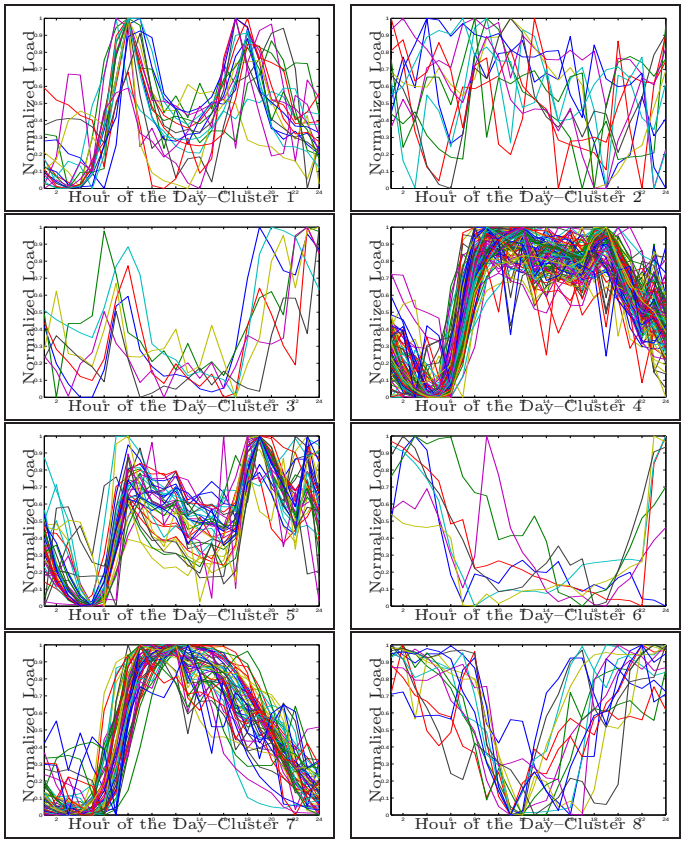

Figure A.3: *Clustering Exercise. Using K-means and the DB validity index it is possible to identify 8 clusters in the set of 245 TDPs.*

Bibliography

[1] L.A. Aguirre, R. Lopes, G. Amaral, and C. Letellier. Constraining the topology of neural networks to ensure dynamics with symmetry properties. *Physical Review E*, 69, 2004.

[2] M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[3] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[4] H. Akaike. Information measures and model selection. *Bulletin of the International Statistical Institute*, 50:277–290, 1983.

[5] G. Altinay. Estimating growth rate in the presence of serially correlated errors. *Applied Economics Letters*, 10(15):967–970, 2003.

[6] N.S. Altman. Kernel smoothing of data with correlated errors. *Journal of the American Statistical Association*, 85:749–759, 1990.

[7] C. Alzate and J.A.K. Suykens. A weighted kernel pca formulation with out-of-sample extensions for spectral clustering methods. Technical Report 06-17, ESAT-SISTA, K.U.Leuven, 2006.

[8] N. Amjady. Short-term hourly load forecasting using time-series modeling with peak load estimation capability. *IEEE Transactions on Power Systems*, 16(4):798–805, 2001.

[9] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software Inc., 2002.

[10] R. Betancourt and H. Kelejian. Lagged endogenous variables and the cochrane-orcutt procedure. *Econometrica*, 49(4):1073–78, 1981.

[11] A. Björkstrom and R. Sundberg. A generalized view on continuum regression. *Scand. Journal of Statistics*, 26:17–30, 1999.

[12] J. Boets, K. De Cock, M. Espinoza, and B. De Moor. Clustering time series, subspace identification and cepstral distances. *Communications in Information and Systems*, 5(1):69–96, 2005.

[13] G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco, 1970.

[14] D. Bunn. Forecasting load and prices in competitive power markets. *Proceedings of the IEEE*, 2(88):163–169, 2000. Invited Paper.

[15] E. Carpaneto, G. Chicco, R. Napoli, and M. Scutariu. Customer classification by means of harmonic representation of distinguishing features. IEEE Bologna Power Tech Conference, 2003.

[16] D. Cochrane and G.H. Orcutt. Application of least-squares regressions to relationships containing autocorrelated error terms. *Journal of the American Statistical Association*, 44:32–61, 1949.

[17] R. Collins. The economics of electricity hedging and a proposed modification for the futures contract for electricity. *IEEE Transactions on Power Systems*, 17(1):100–107, 2002.

[18] P. Crama, J. Schoukens, and R. Pintelon. Generation of enhanced initial estimates for hammerstein systems. *Automatica*, 40:1269–1273, 2004.

[19] N. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, New York, 1993.

[20] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[21] D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227, 1979.

[22] J. De Brabanter. *LS-SVM Regression Modelling and its Applications*. PhD thesis, K.U.Leuven, 2004.

[23] B. De Moor. Daisy: Database for the identification of systems. Department of Electrical Engineering, ESAT-SCD-SISTA, K.U.Leuven, Belgium.

[24] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[25] R.F. Engle, C.W. Granger, J. Rice, and A. Weiss. Semiparametric estimates of the relation between weather and electricity sales. *Journal of the American Statistical Association*, 81(394):310–320, 1986.

[26] R.F. Engle and C.W.J. Granger. Cointegration and error correction: representations, estimation and testing. *Econometrica*, 55:252–276, 1987.

[27] M. Enqvist. Linear Models of Nonlinear FIR Systems with Gaussian Inputs. Technical Report LiTH-ISY-R-2462, Linköping Universitet, Sweden, 2002.

[28] M. Espinoza, B. De Moor, C. Joye, and R. Belmans. Local load analysis with periodic time series and temperature adjustment. In *Proc. of the 15th Power Systems Computation Conference (PSCC) CD-ROM*, 2005.

[29] M. Espinoza, C. Joye, R. Belmans, and B. De Moor. Short term load forecasting, profile identification and customer segmentation: A methodology based on periodic time series. *IEEE Transactions on Power Systems*, 20(3):1622–1630, 2005.

[30] M. Espinoza, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, and B De Moor. A comparative study of LS-SVMs applied to the silverbox identification problem. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

[31] M. Espinoza, J.A.K. Suykens, and B. De Moor. Least squares support vector machines and primal space estimation. In *Proc. of the 42nd IEEE Conference on Decision and Control (CDC)*, pages 5716–5721, 2003.

[32] M. Espinoza, J.A.K. Suykens, and B De Moor. Partially linear models and least squares support vector machines. In *Proc. of the 43rd IEEE Conference on Decision and Control (CDC)*, pages 3388–3393, 2004.

[33] M. Espinoza, J.A.K. Suykens, and B. De Moor. Imposing symmetry in least squares support vector machines regression. In *Proc. of the 44nd IEEE Conference on Decision and Control (CDC)*, pages 5716–5721, 2005.

[34] M. Espinoza, J.A.K. Suykens, and B. De Moor. Kernel based partially linear models and nonlinear identification. *IEEE Transactions on Automatic Control, Special Issue: Linear vs. Nonlinear*, 50(10):1602–1606, 2005.

[35] M. Espinoza, J.A.K. Suykens, and B. De Moor. Load forecasting using fixed-size least squares support vector machines. In J. Cabestany, A. Prieto, and F. Sandoval, editors, *Proceedings of the 8th International Work-Conference on Artificial Neural Networks*, volume 3512 of *Lecture Notes in Computer Science*, pages 1018–1026. Springer-Verlag, 2005.

[36] M. Espinoza, J.A.K. Suykens, and B. De Moor. Short term chaotic time series prediction using symmetric LS-SVM regression. In *Proc. of the 2005 International Symposium on Nonlinear Theory and Applications (NOLTA)*, pages 606–609, Brugge, Belgium, 2005.

[37] M. Espinoza, J.A.K. Suykens, and B. De Moor. Fixed-size least squares support vector machines : A large scale application in electrical load forecasting. *Computational Management Science (Special Issue on Support Vector Machines)*, 3(2):113–129, April 2006.

[38] M. Espinoza, J.A.K. Suykens, and B. De Moor. LS-SVM regression with autocorrelated errors. In *Proc. of the 14th IFAC Symposium on System Identification (SYSID)*, pages 582–587, March 2006.

[39] M. Espinoza, J.A.K. Suykens, and B. De Moor. Structured kernel based modeling: An exploration in short term load forecasting. Technical Report 05-206, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2006.

[40] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2001.

[41] D. Fay, J. Ringwood, M. Condon, and M. Kelly. 24-h electrical load data-a sequential or partitioned time series? *Neurocomputing*, 55:469–498, 2003.

[42] P.H. Franses and R. Paap. *Periodic Time Series Models*. Oxford University Press, 2003.

[43] M. Genton. Classes of kernel for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

[44] M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 10(6):1455–1480, 1998.

[45] I. Goethals, K. Pelckmans, J.A.K. Suykens, and B. De Moor. NARX Identification of Hammerstein Models Using Least Squares Support Vector Machines. Technical Report Internal Report 04-40, ESAT-SISTA, K.U.Leuven, 2004.

[46] C.W.J. Granger and T. Terasvirta. *Modelling Nonlinear Economic Relationships*. Oxford University Press, 1993.

[47] G. Gross and F. Galiana. Short-term load forecasting. *Proceedings of the IEEE*, 75(12):1558–1573, 1987.

[48] R. Guidorzi. *Multivariable System Identification: From Observations to Models*. Bononia University Press, 2003.

[49] G. Guthrie and S Videbeck. High frequency electricity spot price dynamics: An intra-day markets approach. Technical report, New Zealand Institute for the Study of Competition and Regulation, 2002.

[50] E. Haesen, M. Espinoza, B. Pluymers, I. Goethals, V. Thong, J. Driesen, R. Belmans, and B. De Moor. Optimal placement and sizing of distributed generator units using genetic optimization algorithms. *Electrical Power Quality and Utilisation Journal*, 11(1):97–104, 2005.

[51] J. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

[52] W. Härdle. *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge University Press, 1989.

[53] W. Härdle. *Applied Nonparametric Regression*. Cambridge University Press, Cambridge, UK, 1990.

[54] W. Härdle, H. Liang, and J. Gao. *Partially Linear Models*. Physica-Verlag, Heidelberg, 2000.

[55] A. Hardy. On the number of clusters. *Computational Statistics & Data Analysis*, 23:83–96, 1996.

[56] D. Harrison and D.L. Rubinfeld. Hedonic prices and the demand for clean air. *J. Environ. Economics & Management*, 5:81–102, 1978.

[57] S. Haykin. *Neural Networks, A Comprehensive Foundation*. Macmillan, New York, 1994.

[58] S. Heunis and R. Herman. A probabilistic model for residential consumer loads. *IEEE Transactions on Power Systems*, 17(3):621–625, 2002.

[59] H.S. Hippert, D.W. Bunn, and R.C. Souza. Large neural networks for electricity load forecasting: Are they overfitted? *International Journal of Forecasting*, 21:425–434, 2005.

[60] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, and B. De Moor. Primal space sparse kernel partial least squares regression for large-scale problems. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 563–566, Budapest, Hungary, 2004.

[61] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[62] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[63] S-J. Huang and K-R. Shih. Short term load forecasting via ARMA model identification including non-gaussian process considerations. *IEEE Transactions on Power Systems*, 18(2):673–679, 2003.

[64] S. Hylleberg. *Modelling Seasonality*. Oxford University Press, 1992.

[65] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[66] J.A. Jardini, C. Tahan, M. Gouvea, and S.U. Ahn. Daily load profiles for residential, commercial and industrial low voltage consumers. *IEEE Transactions on Power Delivery*, 15(1):375–380, 2000.

[67] J. Johnston. *Econometric Methods*. Economics Series. McGraw-Hill, 1991.

[68] I.T. Jolliffe. *Principal Components Analysis*. Springer Series in Statistics. Springer-Verlag, 1986.

[69] A Juditsky, H. Hjalmarsson, A. Benveniste, B. Deylon, L Ljung, J. Sjöberg, and Q. Zhang. Nonlinear Black-box Modelling in System Identification: mathematical foundations. *Automatica*, 31:1725–1750, 1995.

[70] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. *Transaction American Society Mechanical Engineering, Journal of Basic Engineering*, 83:103–116, 1961.

[71] A. Khotanzad, R. Afkhami-Rohani, and D. Maratukulam. ANNSTLF-artificial neural network short-term load forecaster-generation three. *IEEE Transactions on Power Systems*, 13(4):1413–1422, 1998.

[72] K-H. Kim, H-S. Youn, and Y-C. Kang. Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method. *IEEE Transactions on Power Systems*, 15(2):559–565, 2000.

[73] Judy Klein. *Statistical Visions in Time*. Cambridge University Press, 1997.

[74] R.L. Kosut, G.C. Goodwin, and M.P. Polis. *IEEE Transactions on Automatic Control (Special issue on system identification for robust control design).*, 37(7), 1992.

[75] H. Liao and D. Niebur. Load profile estimation in electric transmission networks using independent component analysis. *IEEE Transactions on Power Systems*, 18(2):707–715, 2003.

[76] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, New Jersey, 1987.

[77] L. Ljung, Q. Zhang, P. Lindskog, and A. Juditski. Estimation of grey box and black box models for non-linear circuit data. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

[78] A.D.P. Lotufo and C.R. Minussi. Electric power systems load forecasting: A survey. Budapest, Hungary, 1999. IEEE Power Tech Conference.

[79] J. Lucia and E.S. Schwartz. Electricity prices and power derivatives: Evidence from the nordic power exchange. *Review of Derivatives Research*, 5(1):5=50, 2002.

[80] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[81] D.J.C. MacKay. Introduction to gaussian processes. In C.M. Bishop, editor, *Neural Networks and Machine Learning*, volume 168 of *Springer NATO-ASI Series F*, pages 133–165, 1998.

[82] D.J.C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11:1035–1068, 1999.

[83] C.L. Mallows. Some comments on C_p. *Technometrics*, 15:661–675, 1973.

[84] E. Mariani and S.S. Murthy. *Advanced Load Dispatch for Power Systems*. Advances in Industrial Control. Springer-Verlag, 1997.

[85] A.I. McLeod. Diagnostic checking of periodic autoregression models with applications. *The Journal of Time Series Analysis*, 15(2):221–223, 1994.

[86] J. McNames, J.A.K. Suykens, and J. Vandewalle. Winning entry of the k.u.leuven time series prediction competition. *International Journal of Bifurcation and Chaos*, 9(8):1485–1500, 1999.

[87] L. Meeus, K. Purchala, and R. Belmans. The belgian power balance. In *Proceedings of the IASTED Internation conference PowerCon, Special Theme: Blackout*, pages 91–96, New York, USA, December 10-12 2003.

[88] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 209:415–446, 1909.

[89] L. Mohan Saini and M. Kumar Soni. Artificial neural network-based peak load forecasting using conjugate gradient methods. *IEEE Transactions on Power Systems*, 17(3):907–912, 2002.

[90] E.A. Nadaraya. On estimating regression. *Theory of Probability and its Application*, 10:186–190, 1964.

[91] J. Nowicka-Zagrajek and R. Weron. Modeling electricity loads in california: ARMA models with hyperbolic noise. *Signal Processing*, 82:1903–1915, 2002.

[92] J. Paduart, G. Horvath, and J. Schoukens. Fast identification of systems with nonlinear feedback. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

[93] K. Pelckmans, M. Espinoza, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Primal-dual monotone kernel regression. *Neural Processing Letters*, 22(2):171–182, 2005.

[94] T. Pena Centeno and N.D. Lawrence. Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis. *Journal of Machine Learning Research*, 7:456–491, 2006.

[95] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78:1481–1497, 1990.

[96] D. Qin and C.L. Gilbert. The error term in the history of time series econometrics. *Econometric Theory*, 17(2):424–50, 2001.

[97] R. Ramanathan, R. Engle, C.W.J. Granger, and C. Vahid-Aragui, F.and Brace. Short-run forecasts of electricity load and peaks. *International Journal of Forecasting*, pages 161–174, 1997.

[98] P.M. Robinson. Root n-consistent Semiparametric Regression. *Econometrica*, 56(4):931–954, 1988.

[99] R. Rosipal and L.J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2:97–123, 2001.

[100] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by backpropagation errors. *Nature*, 323(533-536), 1986.

[101] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings 15th International Conference on Machine Learning (ICML'98)*, pages 515–521, San Francisco, California, 1998. Morgan Kaufmann.

[102] J. Schoukens, G. Nemeth, P. Crama, Y. Rolain, and R. Pintelon. Fast approximate identification of nonlinear systems. *Automatica*, 39(7), 2003.

[103] J. Shawe-Taylor and C.K.I Williams. The stability of kernel principal components analysis and its relation to the process eigenspectrum. *Advances in Neural Information Processing Systems*, 15, 2003.

[104] M. Shoukri, M. Attanasio, and J.M. Sargeant. Parametric versus semi-parametric models for the analysis of correlated survival data: A case study in veterinary epidemiology. *Journal of Applied Statistics*, 25(3):357–374, 1998.

[105] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear Black-box Modelling in System Identification: a Unified Overview. *Automatica*, 31:1691–1724, 1995.

[106] A.J. Smola, T. Friess, and B. Schölkopf. Semiparametric support vector and linear programming machines. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 585–591. MIT Press, 1999.

[107] P. Speckman. Kernel smoothing in partial linear models. *J. R. Statist. Soc. B*, 50:413–136, 1988.

[108] T.P. Speed and B. Yu. Model selection and prediction: normal regression. *Annals of the Institute of Statistical Mathematics*, 45:35–54, 1994.

[109] L. Sranger, J. Schoukens, and G. Horvath. Modelling of a slightly nonlinear system: A neural network approach. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

[110] H. Steinherz, C. Pedreira, and R. Castro. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44–55, 2001.

[111] M. Stone. Comments on model selection criteria of akaike and schwarz. *Journal of the Royal Statistical Society B*, 41:276–278, 1979.

[112] M. Stone and R.J. Brooks. Continuum regression: Cross-validated sequentially contructed prediction ebracing ordinary least squares, partial least squares and principal components regression. *J. R. Statist. Soc. B*, 52:237–269, 1990.

[113] R Sundberg. Continuum regression and ridge regression. *J. R. Statist. Soc. B*, 55:653–659, 1993.

[114] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[115] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor. A support vector machine formulation to PCA analysis and its kernel version. *IEEE Transactions on Neural Networks*, 14(2):447–450, 2003.

[116] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[117] J.A.K. Suykens and J Vandewalle. The K.U.Leuven competition data : a challenge for advanced neural network techniques. In *Proc. of the European Symposium on Artificial Neural Networks (ESANN'2000)*, pages 299–304, Brugges,Belgium, 2000.

[118] J. Taylor and R. Buizza. Neural network load forecasting with weather ensemple predictions. *IEEE Transactions on Power Systems*, 17(2):626–632, 2002.

[119] J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley, 1974.

[120] B.M. Troutman. Some results in periodic autoregressions. *Biometrika*, 66:219–228, 1979.

[121] T. Van Gestel, J.A.K. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle. Predicting financial time series using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks (Special Issue on Financial Engineering)*, 12:809–821, 2001.

[122] T. Van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle. A Bayesian framework for least squares support vector machine classifiers, Gaussian processes and kernel Fisher discriminant analysis. *Neural Computation*, 14:1115–1147, 2002.

[123] T. Van Gestel, B. Baesens, M. Espinoza, J.A.K. Suykens, D. Baestaens, J. Vanthienen, and B. De Moor. Bankruptcy prediction with least squares support vector machines classifiers. In *Proc. of the International Conference on Computational Intelligence for Financial Engineering, CIFER*, pages 1–8, 2003.

[124] T. Van Gestel, J.A.K. Espinoza, M. Suykens, and B. De Moor. Bayesian input selection for nonlinear regression with LS-SVMs. In *Proc. of the 13th System Identification Symposium (SYSID 2003)*, pages 578–583, 2003.

[125] T. Van Gestel, M. Espinoza, B. Baesens, J.A.K. Suykens, C. Brasseur, and B. De Moor. A bayesian nonlinear support vector machine error correction model. *Journal of Forecasting*, 25(2):77–100, March 2006.

[126] T. Van Gestel, J.A.K. Suykens, J. De Brabanter, B. De Moor, and J. Vandewalle. Kernel canonical correlation analysis and least squares support vector machines. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings International Conference on Artificial Neural Networks (ICANN 2001)*, volume 2130 of *Lecture Notes in Computer Science*, pages 381–386, Vienna, Austria, 2001. Springer.

[127] T. Van Herpe, M. Espinoza, B. Pluymers, P. Wouters, F. De Smet, G. Van den Berghe, and De Moor B. Development of a critically ill patient input-output model. In *Proc. of the 14th IFAC Symposium on System Identification (SYSID)*, pages 481–486, March 2006.

[128] T. Van Herpe, B. Pluymers, M. Espinoza, G. Van den Berghe, and B. De Moor. Minimal model for glycemia control in critically ill patients. Technical report, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2006.

[129] V. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

[130] M. Verbeek. *A guide to Modern Econometrics*. Eddison - Wesley, 2000.

[131] V. Verdult. Identification of local linear state space models: the silverbox case study. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

[132] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[133] G. Wahba. Support vector machines, reproducing kernel hilbert spaces, and randomized gacv, chapter 6. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 69–87. MIT Press, 1998.

[134] A.S. Weigend and N.A. Gershenfeld, editors. *Time Series Prediction. Forecasting the Future and Understanding the past*. Addison-Wesley, Reading, MA, 1993.

[135] C.K.I. Williams. Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In M.I. Jordan, editor, *Learning and Inference in Graphical Models*, pages 599–621. Kluwer Academic Press, 1998.

[136] C.K.I. Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. In M.I. Jordan, editor, *Learning and Inference in Graphical Models*, pages 599 – 621. The MIT Press, Cambridge, MA, 1999.

[137] C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In V.Tresp T.Leen, T.Dietterich, editor, *Proc. NIPS 2000*, volume 13, pages 682–688, Vienna, Austria, 2000. MIT press.

[138] C.K.I. Williams and M. Seeger. The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, 2001.

[139] L. Yang and R. Tschernig. Non- and semiparametric identification of seasonal nonlinear autoregression models. *Econometric Theory*, 18:1408–1448, 2002.

[140] G. Young. Maximum likelihood estimation and factor analysis. *Psychometrica*, 6:49–53, 1940.

[141] Y. Yu and W. Lawton. Wavelet based modelling of nonlinear systems. In J.A.K. Suykens and J. Vandewalle, editors, *Nonlinear Modelling: Advanced Black Box Techniques*, pages 119–148. Kluwer Academic Publishers, 1998.

[142] G.U. Yule. On a method of investigating periodicities in disturbed series with special reference to wölfer's sunspot numbers. *Philosophical Transactions Royal Society London Series A*, 226:267–298, 1927.

Curriculum Vitae

Marcelo Espinoza was born in Santiago, Chile, in February 1973. In 1998, he received the degrees of Civil Industrial Engineer and Master of Science in Applied Economics from the Universidad de Chile, with interests in econometrics, time series modeling and business management. Between 1998 and 2001 he worked as a Planning Enginner, and later as Copper Sales Manager, for the chilean company CODELCO. His tasks were related to market research and international contracts management. In 2001 he is accepted for the Master in Artificial Intelligence at the Katholieke Universiteit Leuven, Belgium, in which he graduated magna cum laude. In 2002 he started the Ph.D program in the SCD/SISTA research division of the Electrical Engineering Dept. of the K.U.Leuven, under the supervision of Prof. Dr. Ir. Bart De Moor and Prof. Dr. Ir. Ronnie Belmans.

Publications by the author

Journal Papers

- M. Espinoza, C. Joye, R. Belmans, and B. De Moor. Short term load forecasting, profile identification and customer segmentation: A methodology based on periodic time series. *IEEE Transactions on Power Systems*, 20(3):1622–1630, 2005.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. Kernel based partially linear models and nonlinear identification. *IEEE Transactions on Automatic Control, Special Issue: Linear vs. Nonlinear*, 50(10):1602–1606, 2005.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. Fixed-size least squares support vector machines : A large scale application in electrical load forecasting. *Computational Management Science (Special Issue on Support Vector Machines)*, 3(2):113–129, April 2006.

- J. Boets, K. De Cock, M. Espinoza, and B. De Moor. Clustering time series, subspace identification and cepstral distances. *Communications in Information and Systems*, 5(1):69–96, 2005.

- E. Haesen, M. Espinoza, B. Pluymers, I. Goethals, V. Thong, J. Driesen, R. Belmans, and B. De Moor. Optimal placement and sizing of distributed generator units using genetic optimization algorithms. *Electrical Power Quality and Utilisation Journal*, 11(1):97–104, 2005.

- K. Pelckmans, M. Espinoza, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Primal-dual monotone kernel regression. *Neural Processing Letters*, 22(2):171–182, 2005.

- T. Van Gestel, M. Espinoza, B. Baesens, J.A.K. Suykens, C. Brasseur, and B. De Moor. A bayesian nonlinear support vector machine error correction model. *Journal of Forecasting*, 25(2):77–100, March 2006.

International Conference Papers

- T. Van Gestel, B. Baesens, M. Espinoza, J.A.K. Suykens, D. Baestaens, J. Vanthienen, and B. De Moor. Bankruptcy prediction with least squares support vector machines classifiers. In *Proc. of the International Conference on Computational Intelligence for Financial Engineering, CIFER*, pages 1–8, 2003.

- T. Van Gestel, J.A.K. Espinoza, M. Suykens, and B. De Moor. Bayesian input selection for nonlinear regression with LS-SVMs. In *Proc. of the 13th System Identification Symposium (SYSID 2003)*, pages 578–583, 2003.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. Least squares support vector machines and primal space estimation. In *Proc. of the 42nd IEEE Conference on Decision and Control (CDC)*, pages 5716–5721, 2003.

- M. Espinoza, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, and B De Moor. A comparative study of LS-SVMs applied to the silverbox identification problem. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2004.

- M. Espinoza, J.A.K. Suykens, and B De Moor. Partially linear models and least squares support vector machines. In *Proc. of the 43rd IEEE Conference on Decision and Control (CDC)*, pages 3388–3393, 2004.

- M. Espinoza, B. De Moor, C. Joye, and R. Belmans. Local load analysis with periodic time series and temperature adjustment. In *Proc. of the 15th Power Systems Computation Conference (PSCC) CD-ROM*, 2005.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. Load forecasting using fixed-size least squares support vector machines. In J. Cabestany, A. Prieto, and F. Sandoval, editors, *Proceedings of the 8th International Work-Conference on Artificial Neural Networks*, volume 3512 of *Lecture Notes in Computer Science*, pages 1018–1026. Springer-Verlag, 2005.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. Imposing symmetry in least squares support vector machines regression. In *Proc. of the 44nd IEEE Conference on Decision and Control (CDC)*, pages 5716–5721, 2005.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. Short term chaotic time series prediction using symmetric LS-SVM regression. In *Proc. of the 2005 International Symposium on Nonlinear Theory and Applications (NOLTA)*, pages 606–609, Brugge, Belgium, 2005.

- T. Van Herpe, M. Espinoza, B. Pluymers, P. Wouters, F. De Smet, G. Van den Berghe, and De Moor B. Development of a critically ill patient input-output model. In *Proc. of the 14th IFAC Symposium on System Identification (SYSID)*, pages 481–486, March 2006.

- M. Espinoza, J.A.K. Suykens, and B. De Moor. LS-SVM regression with autocorrelated errors. In *Proc. of the 14th IFAC Symposium on System Identification (SYSID)*, pages 582–587, March 2006.

Internal Reports

- T. Van Gestel, M. Espinoza , J.A.K. Suykens, C. Brasseur, B. De Moor. Bayesian Nonlinear Kernel Based Stock Market Prediction. Internal Report 02-151, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2002.
- M. Espinoza, J.A.K. Suykens, B. De Moor. Structured Kernel Based Modeling: An Exploration in Short Term Load Forecasting. Internal Report 05-206, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2005.
- T. Van Herpe, B. Pluymers, M. Espinoza, G. Van den Berghe, B. De Moor. Minimal model for glycemia control in critically ill patients.Internal Report 06-01, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2006.
- T. Van Herpe, B. Pluymers, M. Espinoza, G. Van den Berghe, B. De Moor.A minimal model for glycemia control in critically ill patients.Internal Report 06-77, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2006.
- M. Espinoza, J.A.K. Suykens, R. Belmans, B. De Moor.Electric Load Forecasting - An application of nonlinear system identification and kernel based modeling.Internal Report 06-84, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2006.