KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Structured Nonlinear System Identification Using Kernel-based Methods

Ricardo Castro-Garcia

Supervisor:
Prof. dr. ir. J.A.K. Suykens

Co-Supervisor:
Prof. dr. ir. J. Schoukens
(Vrije Universiteit Brussel)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2017

Structured Nonlinear System Identification Using Kernel-based Methods

Ricardo CASTRO-GARCIA

Examination committee:
Em. prof. dr. ir. P. Van Houtte, chair
Prof. dr. ir. J.A.K. Suykens, supervisor
Prof. dr. ir. J. Schoukens, co-supervisor
(Vrije Universiteit Brussel)
Prof. dr. ir. B. De Moor
Prof. dr. ir. J. Van Impe
Prof. dr. ir. J. De Brabanter
Prof. dr. ir. E. Reynders
Prof. dr. ir. M. Verhaegen
(Technische Universiteit Delft)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2017

Uitgegeven in eigen beheer, Ricardo Castro-Garcia, Kasteelpark Arenberg 10 - box 2446, B-3001 Leuven (Belgium)

A cat is a puzzle for which there is no solution.
Hazel Nicholson

A mis padres.

Preface

Years ago I gave up. I gave up a life of comfort and bland achievements. Knowing the undefeatable odds I face, I still go forth, for what is life but a continuous struggle against time? In the end, what are we without transcendence?

First I want to thank my supervisor Prof. dr. ir. Johan A.K. Suykens for giving me the chance to do my PhD in his group. Johan, I cannot express the thrill I felt when you decided to offer me this opportunity. Doing a PhD was a dream long postponed for me and coming to work with you was a marvelous way to finally fulfill it. During these last years I have learned a lot, some times with sudden epiphanies and some others painstakingly slowly. I am keenly aware that all this process came to happen in the first place because you decided to give me a chance. Thank you.

I also want to thank my co-supervisor Prof. dr. ir. Johan Schoukens for his teachings. First, he gave me the great opportunity to be at VUB's *Measuring, Modelling and Simulation of (Non)linear Dynamic Systems* summer school which allowed me to meet many great researchers and gave me many useful tools for my research. Secondly, his opportune observations led me to improve my research and find new paths enabling me to do better contributions to the field of system identification.

I thank the jury members for their time and dedication to help me improving this work: Prof. dr. ir. Paul Van Hautte, chairman of the jury, Prof. dr. ir. Edwin Reynders, secretary of the jury, Prof. dr. ir. Bart De Moor, Prof. dr. ir. Joseph De Brabanter, Prof. dr. ir. Jan Van Impe and external jury member Prof. dr. ir. Michel Verhaegen.

Many people have crossed my path during these years, but some of the most influential persons in any researcher's career are his colleagues. I worked with some of them directly and, with some others, shared moments and experiences. First Rocco, Raghvendra, Vilen, Marko, Marco, Gervasio, Xiaolin and Siamak. Later, also Yuning, Yunlong, Michael, Bertrand, Emanuele, Carlos, Ángela and Hans. I came to have other colleagues that even if not part of our particular group, were there to share a coffee or a laugh: Puya, Edward, Andreas, Jasper, Reza, Federica, Fernando, Pantelis and Domagoj. Thank you all for being there and making this journey an even better experience.

Also, there are people with whom you share great moments, who you trust and know you can rely on. Tania, Valentina, Tim and Camila: For all of those great moments we shared, thanks! Beside this, I have to specially thank Maritza: your hospitality and generosity, your good humor and warm hearth made my time in Leuven a much better experience. Sergio, we started this trip together many years ago, way before coming to Leuven. Many things have changed since then and it is impossible to even try to list everything I feel grateful to you for. I wish I have the chance in my lifetime to repay you in kind. Thank you brother.

Of course, there are people that influence your life in very particular ways. Carolina, your guidance was crucial for me before beginning my PhD. I can safely say that were it not for you, I would have not ended up working in this particular group under Johan's supervision. It is because of you that I took this path. Thank you for that. Mauricio, it was you who provided those additional input and feedback that every researcher needs. This journey would have been very difficult without your help. I greatly appreciate all the work we did together and all the things I was able to learn from you, thank you.

Alex, we had fun and argued about serious things. You were there when it seemed that I was completely on my own, thank you. Koen, thank you for your patience and diligence. Your opportune and precise contributions greatly helped to improve my research. I feel fortunate I was able to work with you.

Giulio, your way of seeing things is inspiring. I have learned a lot from you beyond the academic world. I keep many good memories and am glad we could work together.

Lynn, I hope you are aware of how much I value your friendship and how much I trust you. We have shared so many good moments in these few years that it feels like we have always been friends. Thank you for listening to me, for being there to have a beer or three from time to time, for demonstrating over and over again what true friendship means.

Zahra, I am happy I had the opportunity to be your friend for so many years now. What better office mate could I have wished for than a true friend like you? Thanks for your (exceptional) patience with me, all the support you gave me and all the wisdom you shared with me. Thank you for standing my questions and endless explanations. I cannot imagine feeling more at ease working with someone else. I will miss you dearly.

Lol, gracias por el apoyo incondicional que me has dado, por estar ahí, por ser paciente. A veces es necesario tener a alguien que crea en uno, especialmente cuando se siente que nadie más, ni siquiera uno mismo, lo hace. Gracias por ser esa persona, por darme esperanza cuando todo se veía oscuro, por ayudarme a soñar con mejores cosas, por ni siquiera considerar que pudiera fallar. Gracias por tu inocencia y sencillez. Gracias por tu apoyo incondicional.

Familia, el apoyo que me han brindado a través de los años ha sido fundamental para mí. Gracias a que ustedes creyeron en mí he podido superar muchos obstáculos. Gracias a que sé que están ahí, tengo la confianza de que tengo a dónde regresar. Mil gracias.

Leo y Liliana, siempre voy a tener una deuda de gratitud con ustedes. En todo momento tengo presente en mi vida la increíble generosidad que tuvieron conmigo y toda la confianza que me demostraron. Todo su apoyo y amor ha sido una pieza fundamental para llegar a ser la persona que soy. Ojalá la vida me permita algún día ayudar a otras personas siguiendo el maravilloso ejemplo de vida que ustedes me han dado.

Papá y mamá, este logro es para ustedes que son mi ejemplo y mi motivación, a quienes todo debo. Los amo profundamente y les agradezco todo lo que han hecho por mí. Agradezco su sacrificio, sus enseñanzas y sobre todo su ejemplo de vida. Me siento muy orgulloso de poder decir que personas tan maravillosas como ustedes son mis padres y pensar en ello hace que me llene de gratitud. Papá, te admiro muchísimo y guardo en la memoria muchos momentos en que nuestras charlas marcaron mi vida. Mamá, tú me enseñaste a abrir mi mente, a ser crítico, a respetar la vida. Gracias a ti amo la ciencia.

Abstract

The development of models is a crucial part of modern science. Our comprehension of the different phenomenons in all of its fields is directly related to the models we create. These models allow obtaining new insights and predicting the outcome of new experiments. This thesis focuses on the development of techniques for the identification of block-oriented nonlinear (BONL) models. These models have been shown in the past to be flexible and powerful for describing a multitude of phenomenons and have received a lot of attention in the scientific community during the last few decades. This means that the task of finding new and better alternatives than those currently available is a difficult one. To undertake it, we use Least Squares Support Vector Machines (LS-SVM) as our base method.

The thesis is divided in four parts and in each one different methods are presented. Part I focuses on merging the best linear approximation (BLA) with LS-SVM for the identification of Hammerstein and Wiener systems. Each of these techniques is used where it excels: BLA is used for the identification of the linear blocks while LS-SVM is used for modeling the static nonlinearities. In this part three methods are presented: The first offers a way to use the inversion of a (previously) identified linear block for the identification of Hammerstein systems in the presence of measurement noise. The other two methods are for the identification of Hammerstein and Wiener systems respectively and offer a reformulation of LS-SVM where information of the system, given by the BLA, is incorporated.

In Part II a new approach for the identification of Hammerstein and Wiener systems is presented. This approach relies on the extraction of information from the system based on its behavior during steady state. A method for identifying Single-Input Single-Output (SISO) Hammerstein systems is presented and then extended to the Multiple-Input Multiple-Output (MIMO) case. A third method is presented for the identification of SISO Wiener systems. This method offers three different alternatives for such identification (i.e. two parametric and one non-parametric).

In Part III a new methodology for the identification of Hammerstein systems is offered.

The method takes advantage of the Hammerstein system structure as the impulse response of such systems allows the identification of their dynamic blocks. First the SISO case is presented and then it is extended to the MIMO one.

Finally, in Part IV two methods focusing on machine learning are presented. The first of these methods focuses on the complexity reduction of fixed size schemes for black box modeling. The second method introduces a way to include frequency domain information into the LS-SVM model. It is shown that this methodology can offer better results for the modeling of dynamical systems than NARX LS-SVM.

Abstract

Het ontwerpen van modellen is een cruciaal onderdeel van de moderne wetenschappen. Ons begrip van de verschillende fenomenen in all de gebieden, is sterk gerelateerd aan de modellen die we creëren. Deze modellen kunnen nieuwe inzichten omtrent het fenomeen brengen en het resultaat van nieuwe experimenten voorspellen. Deze thesis focust op de ontwikkeling van nieuwe technieken voor de identificatie van bloksgewijze non-lineaire (BONL) modellen. Deze modellen hebben in het verleden reeds aangetoond flexibel en krachtig te zijn voor het omschrijven van meerdere fenomenen en hebben reeds veel aandacht gekregen in de wetenschappelijke wereld in de laatste decennia, waardoor er al krachtige technieken voor identificatie bestaan. Dit betekent dat het vinden van nieuwe en betere alternatieven dan de reeds bestaande technieken geen gemakkelijk taak is. Om deze taak aan te pakken hebben we gebruik gemaakt van krachtige technieken uit de machine learning wereld. Meer bepaald gebruiken we least squares support vector machines (LS-SVM) als onze basis methode.

Deze thesis is onderverdeeld in vier delen, waarin in elk deel een verschillende methode wordt voorgesteld. Deel I focust op het samenbrengen van de best linear approximation (BLA) methode met LS-SVM voor het identificeren van Hammerstein en Wiener systemen. Elk van deze technieken wordt gebruikt waar ze het best functioneren: BLA wordt gebruikt voor de identificatie van de lineaire blokken, terwijl LS-SVM gebruikt wordt voor het modelleren van de statische non-lineaire delen. In dit deel worden drie methoden voorgesteld: de eerste presenteert een manier om de inversie van een (eerder) geïdentificeerd lineair blok te gebruiken voor het identificeren van een Hammerstein systeem in het geval van ruis. De andere twee methoden zijn ontworpen voor het identificeren van Hammerstein en Weiner systemen respectievelijk, en bieden een herformulering van LS-SVM aan waarbij informatie over het systeem, gegeven door de BLA, is geïncorporeerd.

In deel II wordt een nieuwe manier voor het identificeren van Hammerstein of Wiener systemen voorgesteld. Deze aanpak komt voort uit de extractie van informatie van het systeem, vanuit het gedrag van dat systeem tijdens een stabiele status. Een methode voor het identificeren van Single-Input Single-Output (SISO) Hammerstein systemen is voorgesteld en dan uitgebreid naar het Multiple-Input Multiple-Output (MIMO) geval. Een derde methode is voorgesteld voor de identificatie van SISO Wiener systemen. Deze methode biedt drie verschillende alternatieven aan voor de identificatie (namelijk, twee parametrische en één niet-parametrische).

In deel III een nieuwe methodologie voor het identificeren van Hammerstein systemen is voorgesteld. Deze methode gebruikt de Hammerstein structuur, aangezien het impuls resultaat dit systeem de identificatie van de dynamische blokken toelaat. Eerst wordt het SISO geval besproken en dit wordt daarna uitgebreid naar het MIMO geval.

Uiteindelijk stelt deel IV twee methoden die focussen op machine learning voor. De eerste methode focust op de complexiteit reductie van vaste grootte schema's voor black box modelleren. De tweede methode introduceert een manier om het frequentie domein te introduceren in het LS-SVM model. Er wordt aangetoond dat deze methodologie betere resultaten voor het modelleren van dynamische systemen kan bekomen dan het NARX LS-SVM model.

Abbreviations

%MAE	Normalized mean absolute error
ARX	Auto regressive model with exogenous inputs
BLA	Best linear approximation
BONL	Block oriented nonlinear
CSA	Coupled simulated annealing
DFT	Discrete Fourier transform
EDF	Effective degrees of freedom
EMF	Electromotive force
FD-LSSVM	Frequency division least squares support vector machines
FIR	Finite impulse response
FRF	Frequency response function
FS-LSSVM	Fixed size least squares support vector machines
FS-OLS	Fixed size ordinary least squares
FS-RR	Fixed size ridge regression
GA	Genetic algorithm
GP	Gaussian process
KKT	Karush–Kuhn–Tucker
LS	Least squares
LS-SVM	Least squares support vector machines
LTI	Linear time invariant
LTV	Linear time varying
MAE	Mean absolute error
MIMO	Multiple-input multiple-output
MLPRS	Multi level pseudo random signal
NARMAX	Nonlinear autoregressive moving average model with exogenous inputs
NARX	Nonlinear autoregressive exogenous model
OE	Output error
PEM	Prediction error method
PISPO	Period in same period out
PRBS	Pseudo random binary signal

QP	Quadratic programing
RBF	Radial basis function
RKHS	Reproducing kernel Hilbert space
RMS	Root mean square
RMSE	Root mean square error
RR	Ridge regression
SA	Simulated annealing
SISO	Single-input single-output
SNR	Signal to noise ratio
SURE	Stein’s unbiased risk estimate
SVD	Singular value decomposition
SVM	Support vector machine

Nomenclature

$\varphi(\cdot)$	Mapping function to a feature space (feature map)
x	Scalar
$\boldsymbol{x}$	Vector
$\boldsymbol{X}$	Matrix
$x(t)$	Time domain signal
$X(k)$	Frequency domain signal
$\boldsymbol{X}^\top$	Transpose of the matrix $\boldsymbol{X}$
$\boldsymbol{x}_i$	ith element of the vector $\boldsymbol{x}$
$\boldsymbol{X}_{ij}$	ijth element of the matrix $\boldsymbol{X}$
$f(\cdot)$	Function
$\mathbf{1}_N$	Column vector of ones of length N
$\min\limits_{x} f(x)$	Minimization over x. The minimum of $f(x)$ is returned
$\arg\min\limits_{\boldsymbol{x}} f(x)$	Minimization over x. Optimal x is returned
$\lvert\cdot\rvert$	Absolute value
$\lVert\cdot\rVert$	L_2 norm
ω	Angular frequency
ϕ	Phase
$\delta(t)$	Kronecker delta function
$E\left\{\cdot\right\}$	Expectation operator

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research motivation

The development and analysis of models occupies an important role at the very core of modern science. From physics to engineering, almost all fields of science require representations of different phenomenons. These representations are commonly known as models and they allow to obtain new insights concerning the modeled phenomenon and to predict the outcome of experiments involving it (Eskinat, Johnson, & Luyben, 1991). It is the case that nonlinear models, even simple ones, often result in better approximations to process dynamics than linear ones. A considerable amount of research has been carried out in the last decades in the field of nonlinear system identification. A popular approach is to employ one of the block structured nonlinear models introduced in the literature where systems are represented as interconnected linear and nonlinear blocks (Billings & Fakhouri, 1982).

In computer science, machine learning is a subfield aiming to give computers the ability to learn without being explicitly programmed. Having evolved from the study of pattern recognition, it deals with algorithms that can learn from (and make predictions on) data (Ron & Foster, 1998) and in this sense can make data-driven predictions or decisions. In the field of machine learning, kernel methods are a class of algorithms for pattern analysis where the most iconic characteristic is the use of kernel functions allowing a cheap operation of the algorithms in high dimensional spaces. This in turn allows nonlinear problems to be solved using linear formulations. In this thesis the base for most of the presented methods will be Least Squares Support Vector Machines (LS-SVM) (Suykens et al., 2002).

It is interesting to note that there are many different model structures in the system identification literature e.g. ARX, Output-Error methods, Box-Jenkins, state space, block oriented models, etc. In addition, different parametrizations can be used e.g. linear, polynomial, piecewise linear, etc. For nonlinear system identification, support vector machines and kernel methods have been successfully applied in the past for certain classes of model structures. In general, the different options for model structure and parametrization can be used when implementing kernel methods in their primal representation. However, the situation changes when working in the dual as the resulting models are no longer parametric. Therefore, it is challenging to incorporate prior knowledge about the structure of the system within a primal-dual optimization setting of kernel methods. Given this difficulty, and considering the intrinsic advantages of the dual representation, it becomes clear that this is an important and interesting challenge. The aim of this research is to advance in this area which is at the interface between nonlinear system identification and machine learning by combining and integrating the best of both paradigms and employing both parametric and kernel-based approaches with suitable regularization schemes.

1.2 Research objectives

In this section we outline the general objectives of this research.

Propose new techniques for incorporating prior knowledge about the structure of the system into black-box modeling schemes

Methods like LS-SVM are inherently of a black box nature, which means that the models produced are employed without reference to a physical background. Sometimes, however, in addition to the input and output data some additional information about the underlying system is available. An example of this in the identification of block structured nonlinear dynamical systems is when information about the underlying structure of the system is available. Even though not using such additional information is a waste, when using black box methods that is what happens as normally there is no way to include it into the model. Clearly then, finding ways to incorporate this prior knowledge into the model constitutes a natural improvement for methods that otherwise would ignore this information.

Apply and compare the methods in system identification benchmarks and novel applications

System identification is a mature branch of science that has received a lot of attention in the last decades. This means that many powerful methods for system identification are currently available. Despite this, new methodologies keep appearing on a daily basis indicating that there is still room for improvement. Additionally, several data sets exist in the system identification community allowing a fair comparison.

Go beyond the existing class of model structures in nonlinear system identification that can be currently handled with kernel methods

Single-input single output (SISO) block structured nonlinear systems like the Wiener or the Hammerstein systems have been thoroughly studied in the literature. However, their multiple input multiple output (MIMO) counterparts have received much less attention due to their inherently more complicated nature. However, systems like these represent a very important part of the applications where system identification techniques would be useful. This means that techniques that can deal with this type of problems (e.g. MIMO structures) should receive more attention to improve the reach of the system identification community.

1.3 System Identification

In system identification the aim is to build mathematical models based on observed data from the system. Clearly this puts system identification at the core of the scientific method. Loosely defined, a system is an entity which can be affected by external stimuli and through the interaction of different variables, observable signals are produced (i.e. outputs). The external stimuli that can be separated in two categories: Those that can be manipulated by the observer are called inputs while those that cannot are called disturbances. Also disturbances can be separated into those that can be directly measured and those that are only noticed indirectly through their effect in the output.

When interacting with a system, an idea of how its variables relate to each other is called a model of the system. Note that the use of mathematical models is common to all fields of engineering and physics. Models are constructed from observed data and to do so there are two main possibilities, namely modeling and system identification. In modeling the system is split into subsystems with well understood properties relying on earlier empirical work. These subsystems are then merged mathematically to obtain a model of the whole system. System identification, on the other hand, is directly based on experimentation where inputs and outputs are measured and subjected to data analysis in order to infer a model (Ljung, 1999). The latter is the central topic of this thesis and in particular, the focus will be on block oriented nonlinear (BONL) system identification.

1.4 Block Structured System Identification

Block oriented nonlinear (BONL) models consist of the interaction of linear time-invariant (LTI) dynamical blocks and static nonlinear elements that can be connected in many different ways (e.g. series, parallel, feedback,etc.). The blocks themselves can be represented in different forms. For instance the LTI blocks can be parametric (e.g. transfer functions or state space representations) or non parametric (e.g. impulse response or frequency response). Similarly, the nonlinear blocks can be parametric, nonparametric, with memory or memoryless. All these different available options point towards the high flexibility of BONL models which allows the capture of a wide variety class of complex and nonlinear systems. This in part explains why BONL has received so much attention in the last decades (Giri & Bai, 2010).

Note that even though the BONL models are powerful tools for representing nonlinear dynamical systems, the blocks used do not generally correspond to physical components. This means that the intermediate variables between them are generally artificial and usually cannot be physically measured. It is only natural then that the combination of the dynamics, nonlinearities and the impossibility to measure intermediate variables renders the problem of estimating such models into a difficult one. This also explains why the main focus of attention in the research of BONL models is mainly on simpler structures.

Even though commonly the dynamics of the system can be approximated by a linear system, often it is the case that there are static nonlinearities at the input or output. Although many different structures exist, in this thesis the focus is on specific BONL structures, namely the Hammerstein and Wiener systems. These structures are composed by two blocks in series as shown in Figs. 1.1 and 1.2 respectively. Typical examples are actuators being nonlinear or sensors having nonlinear characteristics. It is usually considered that Hammerstein systems contain static nonlinearities at its input and, similarly, Wiener systems have static nonlinearities at its output (Ljung, 1999).

In this work, the q-notation, which is frequently used in system identification literature and software, will be employed. The operator q is a time shift operator of the form $q^{-1}x(t) = x(t-1)$.

1.4.1 Hammerstein Systems

Hammerstein systems were introduced by the German mathematician A. Hammerstein in 1930 (Hammerstein, 1930). As shown in Fig. 1.1 the input of the system first goes through a static nonlinear block and the resulting output passes then through an LTI block. In the first block, all the nonlinearities of the system are accounted for, while the second block describes all the dynamics of the system. Commonly, the Hammerstein

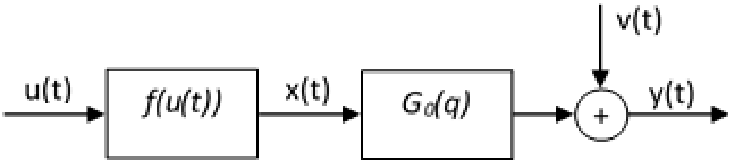


Figure 1.1: A Hammerstein system. $G_0(q)$ is a linear dynamical system and $f(u(t))$ is a static nonlinearity and $v(t)$ represents the measurement noise.

structure is used to model systems where the static nonlinear is at the input of the system.

Although it is a simple structure, Hammerstein systems can accurately describe many nonlinear systems and has been used in areas like control (Fruzzetti, Palazoglu, & McDonald, 1997), biological processes (Hunter & Korenberg, 1986), signal processing (Stapleton & Bass, 1985), chemical processes (Eskinat et al., 1991), electrically stimulated muscles (Hunt, Munih, Donaldson, & Barr, 1998), power amplifiers (Kim & Konstantinou, 2001), electrical drives (Balestrino, Landi, Ould-Zmirli, & Sani, 2001), thermal microsystems (Sung, 2002), physiological systems (Dempsey & Westwick, 2004), sticky control valves (Srinivasan, Rengaswamy, Narasimhan, & Miller, 2005), solid oxide fuel cells (Jurado, 2006), and magneto-rheological dampers (J. Wang, Sano, Chen, & Huang, 2009).

There are many methods for Hammerstein system identification in the literature. With so many approaches available, it is natural that many different ways of classifying them exist. Some of these possible classifications are (M. Schoukens & Tiels, 2016): Kernel-based and mixed parametric-nonparametric identification algorithms (Hasiewicz, Mzyk, Śliwiński, & Wachel, 2012; Mzyk, 2014; Risuleo, Bottegal, & Hjalmarsson, 2015), parametric approaches (Chang & Luus, 1971; Crama & Schoukens, 2004; J. Schoukens, Widanage, Godfrey, & Pintelon, 2007), overparametrization (Bai, 1998; Falck, Suykens, Schoukens, & De Moor, 2010; Risuleo et al., 2015), blind identification (Bai, 2002; Vanbeylen, Pintelon, & Schoukens, 2008). Some methods for MIMO Hammerstein system identification are presented in Goethals, Pelckmans, Suykens, and De Moor (2005); Gomez and Baeyens (2004); Jeng and Huang (2008); Lee, Sung, Park, and Park (2004); Verhaegen and Westwick (1996) and Al-Duwaish and Karim (1997). Hammerstein structures containing dynamic backlask or hysteresis are considered in Giri, Rochdi, Brouri, and Chaoui (2011) and Z. Wang, Zhang, Mao, and Zhou (2012).

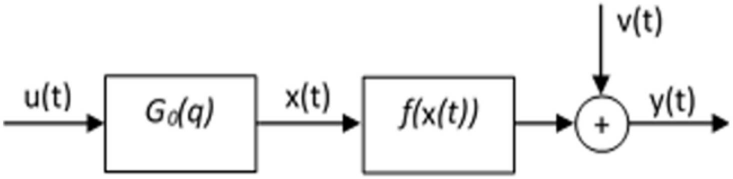


Figure 1.2: A Wiener system. $G_0(q)$ is a linear dynamical system, $f(x(t))$ is a static nonlinearity and $v(t)$ represents the measurement noise.

1.4.2 Wiener Systems

In 1958 N. Wiener studied a model where the input went through an LTI block and the resulting output then went through a nonlinear block (Wiener, 1958). This is known as the Wiener system and it is shown in Fig. 1.2. In Wiener systems, the nonlinear block can represent for instance sensor nonlinearities or nonlinear effects at the output of the system. Examples of this include overflow valves and limit switch devices in mechanical systems.

Wiener models are known to be able to approximate a general class of nonlinear systems with an arbitrarily high accuracy under the assumption of fading memory (Boyd & Chua, 1985), a theoretical fact checked in practice in many practical applications like chemical processes (Kalafatis, Wang, & Cluett, 2005; Zhu, 1999), biological systems (Hunter & Korenberg, 1986) and others.

As in the Hammerstein case, many different classifications exist for Wiener systems according to their properties (M. Schoukens & Tiels, 2016): Nonparametric or semi-parametric (Greblicki, 1992, 1997; Hasiewicz et al., 2012; Lindsten, Schön, & Jordan, 2013; Mzyk, 2007, 2014; Wachel & Mzyk, 2016), parametric approaches (Billings & Fakhouri, 1977; Crama & Schoukens, 2001a; Hunter & Korenberg, 1986; D. T. Westwick & Kearney, 2003; Wigren, 1993), minimal Lipschitz (Pelckmans, 2011), (orthogonal) basis function expansion (Aljamaan, Westwick, & Foley, 2014; Lacy & Bernstein, 2003), blind identification algorithms (Vanbeylen, Pintelon, & Schoukens, 2009), recursive approach (Greblicki, 2001; Wigren, 1993) separable least-squares (Bruls, Chou, Haverkamp, & Verhaegen, 1999), subspace-based methods (D. Westwick & Verhaegen, 1996). The MIMO Wiener system case is considered in Janczak (2007) and D. Westwick and Verhaegen (1996). In Giri, Radouane, Brouri, and Chaoui (2014) systems that contain backslash nonlinearities are considered.

Most of the methodologies in the literature consider that the noise source is present

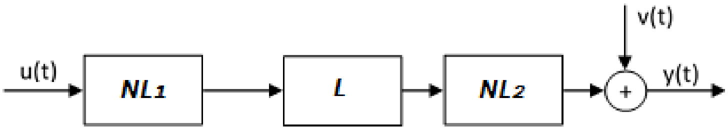


Figure 1.3: A Hammerstein-Wiener system. NL_1 and NL_2 are static nonlinearities while L is a linear dynamical system. $v(t)$ represents the measurement noise.

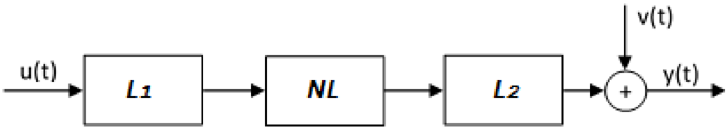


Figure 1.4: A Wiener-Hammerstein system. L_1 and L_2 are linear dynamical systems while NL is a static nonlinearity. $v(t)$ represents the measurement noise.

at the output of the system only. However, some methods allow for process noise between the linear and nonlinear blocks (Hagenblad, Ljung, & Wills, 2008; Lindsten et al., 2013; Wahlberg, Welsh, & Ljung, 2014, 2015).

1.4.3 Others

When a Hammerstein model is followed in series by a Wiener one a new model structure called the Hammerstein-Wiener system arises (see Fig. 1.3). In a similar way, when a Wiener system is followed by a Hammerstein one the resulting system is referred to as a Wiener–Hammerstein structure (see Fig. 1.4). These new structures offer higher modeling capabilities. For example the Hammerstein–Wiener model is more convenient when both actuator and sensor nonlinearities are present. Also, it has been successfully applied in the modeling of several physical processes like polymerase reactors (Lee et al., 2004), ionospheric processes (Palanthandalam-Madapusi, Ridley, & Bernstein, 2005), PH processes (Park, Sung, & Lee, 2006), etc. When feedback phenomena are involved, closed-loop model structures can also be used and when modeling multichannel topology systems like electric power distribution, communication nets, multi-cell parallel power converters, etc. parallel block oriented models are useful.

1.5 Experiment Design

The construction of a model consists of three stages: first a data set is constructed from observed data, second a set of candidate models is selected (i.e. a model structure), finally a rule to assess the models is chosen (Ljung, 1999):

- Sometimes the data are recorded during a specifically designed identification experiment where the user determines the signals to measure, when to measure and even the input signals. The objective of the experiment design is then to make these choices so that the data obtained is maximally informative. The vast majority of the methods presented in this thesis fall into this category. In other occasions the user must work with data from the normal operation of the system.
- The a priori knowledge about the system plays a crucial role for selecting the set of models from which one will be finally chosen. Models that are employed without reference to physical background (i.e. the parameters do not reflect physical considerations in the system) are called black box models. When the models mix adjustable parameters with physical interpretable ones they are called gray box models.
- To determine the best model of the selected set is the task of the identification method. Usually, to assess the quality of a model, metrics based on how well the model can reproduce measured data are used.

After performing the steps mentioned above the model should be validated, if the model turns out to be deficient, it is then rejected and the process must re-start. A representation of this process can be found in Fig. 1.5.

1.6 Kernel methods

Kernel methods, as used in the area of support vector machines, usually can be described in two steps. First there is a mapping of the inputs into a higher dimensional feature space. This is, images of the inputs are obtained into the higher dimensional space. Second, there is a learning algorithm in charge of discovering linear patterns in that space. Bear in mind that the research in statistics and machine learning about detecting linear relations has been going on for decades. This means that the knowledge in this area is robust and well understood. Also, it is possible to represent linear patterns efficiently in high dimensional spaces through the use of kernel functions.

A Mercer kernel is a function k that for all $\boldsymbol{x}, \boldsymbol{z} \in \boldsymbol{X}$, with $\boldsymbol{X} \in \mathbb{R}^n$, satisfies $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{z}) \rangle$ where $\varphi(\cdot)$ is a mapping from $\boldsymbol{X}$ to an (inner product) feature

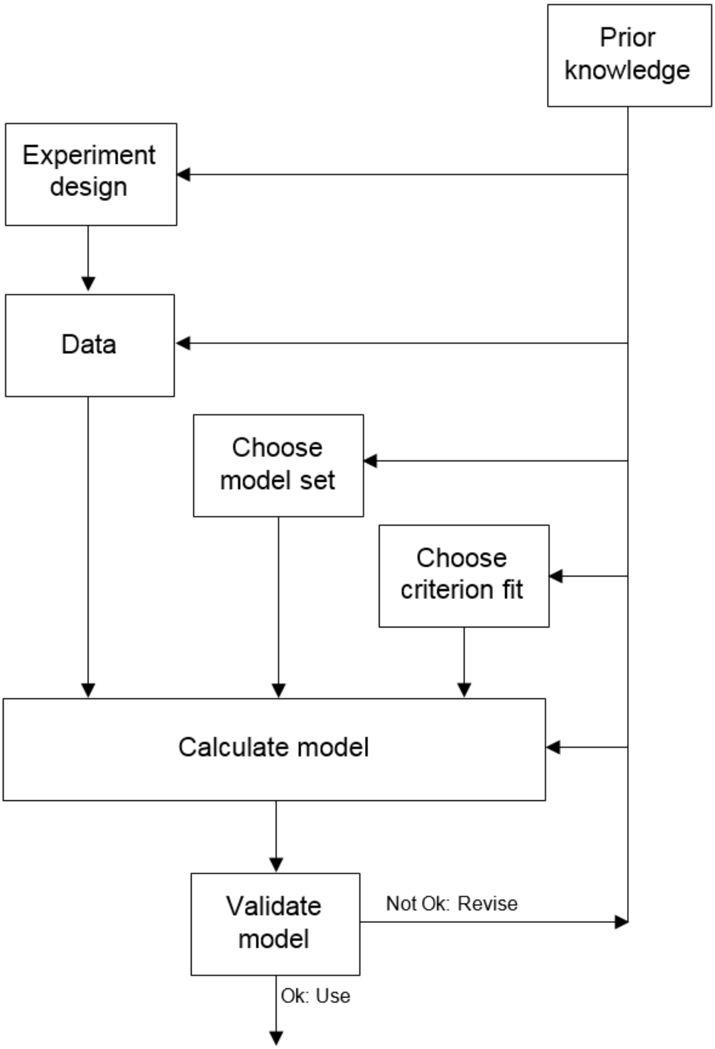


Figure 1.5: System identification loop (Ljung, 1999).

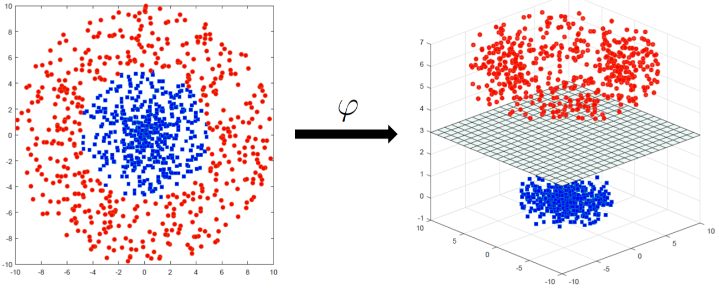


Figure 1.6: The function φ takes the input data into a feature space where the nonlinear (separation) pattern appears linear.

space F (Shawe-Taylor & Cristianini, 2004):

$$\varphi : \boldsymbol{x} \to \varphi(\boldsymbol{x}) \in F. \tag{1.1}$$

The embedding of the data into the vector space called the feature space is fundamental in kernel methods. In this space linear relations are sought among the images of the data. Interestingly the coordinates of the embedded points are not necessary, only their pairwise inner products are required thanks to the way the algorithms are implemented. Also, the pairwise inner products can be computed directly from the original data items in an efficient way by using a kernel function. All of these elements guarantee that even though the used algorithms are meant for linear functions, nonlinear relations in data can be discovered through the use of nonlinear embedding mappings.

In Fig. 1.6 an illustration of these concepts is presented where two classes that cannot be linearly separated in the original input space are shown. However, when φ is used, the data is taken to a higher dimensional space where they can be clearly separated by a plane.

Several kernel methods have been introduced in the literature. In the following sections some of the most commonly used methods will be briefly revised.

1.6.1 Kernel Ridge Regression

Let us consider the problem of finding a function $y = g(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ that best interpolates a given training set $\mathcal{S}$ such that $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$ with N the number of samples, $\boldsymbol{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, $\boldsymbol{w} \in \mathbb{R}^n$ and $i = 1, \ldots, N$. This task is commonly known as linear interpolation (i.e. fitting a hyperplane through the given n-dimensional points). It is usually the case that solutions that minimize the error while keeping the norm of $\boldsymbol{w}$ small are preferred (Shawe-Taylor & Cristianini, 2004).

Let us define $\xi_i = y_i - g(\boldsymbol{x}_i)$ and therefore $\boldsymbol{\xi} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}$ with $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]^\top \in \mathbb{R}^N$, $\boldsymbol{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times n}$ and $\boldsymbol{w} = [w_1, \ldots, w_n]^\top \in \mathbb{R}^n$. A loss function can be then defined as

$$J(\boldsymbol{w}, \mathcal{S}) = \|\boldsymbol{\xi}\|_2^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}). \tag{1.2}$$

This finally leads to an estimation of $\boldsymbol{w}$ as

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}. \tag{1.3}$$

This is commonly known as the primal representation of least squares. If the inverse of $\boldsymbol{X}^\top\boldsymbol{X}$ exists, $\hat{\boldsymbol{w}}$ can also be expressed as

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y} = (\boldsymbol{X}^\top\boldsymbol{X})(\boldsymbol{X}^\top\boldsymbol{X})^{-2}\boldsymbol{X}^\top\boldsymbol{y} = \boldsymbol{X}^\top\boldsymbol{\alpha}, \tag{1.4}$$

which is known as the dual representation (Shawe-Taylor & Cristianini, 2004). Note that in the dual representation $\hat{\boldsymbol{w}}$ appears as a linear combination of the training points, i.e. $\hat{\boldsymbol{w}} = \sum_{i=1}^N \alpha_i\boldsymbol{x}_i$.

It is possible to look for a tradeoff between the size of the norm of $\hat{\boldsymbol{w}}$ and the error $\boldsymbol{\xi}$. This is what is known as ridge regression. Ridge regression modifies then Least Squares by restating the objective function as

$$J(\boldsymbol{w}, \mathcal{S}) = \lambda\|\boldsymbol{w}\|^2 + \sum_{i=1}^N (y_i - \boldsymbol{w}^\top\boldsymbol{x}_i)^2, \tag{1.5}$$

with λ a fixed positive constant. Note that $\lambda = 0$ is allowed and this means that Least Squares is a special case of Ridge Regression (Saunders, Gammerman, & Vovk, 1998).

Similarly as in the least squares case, ridge regression can be expressed either in the primal with

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}, \tag{1.6}$$

or the dual with

$$\hat{\boldsymbol{w}} = \boldsymbol{X}^\top\boldsymbol{\alpha}, \tag{1.7}$$

and

$$\boldsymbol{\alpha} = (\boldsymbol{X}\boldsymbol{X}^\top + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}. \tag{1.8}$$

It is fundamental to note that in the dual, the information from the training examples is given by the inner products between pairs of training points. The matrix $\boldsymbol{X}\boldsymbol{X}^\top$ is usually referred to as the Gram matrix. Also, note that the derivations of the dual correspond to the introduction of Lagrange multipliers in a constrained optimization problem, i.e. the minimization of $J(\boldsymbol{w}, \mathcal{S})$ such that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\xi}$ (Saunders et al., 1998).

So far only the linear case has been considered for Ridge Regression. If a linear regression is desired in some feature map, first the mapping from the original space to a higher dimensional space F has to be chosen, i.e. $\varphi : \boldsymbol{X} \to F$. To apply this, it is possible to define a kernel function k corresponding to the dot product $\varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j)$ with $\varphi(\boldsymbol{x}) \in \mathbb{R}^{n_h}$. With this, it is not necessary to explicitly know $\varphi(\cdot)$ as long as $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j)$ is known. The choosing of $k(\cdot,\cdot)$ can be done thanks to Mercer's theorem and is addressed in V. N. Vapnik (1998).

Note that with the use of $k(\cdot,\cdot)$, in the dual formulation the estimation of the output for new points $\boldsymbol{X}_*$ becomes

$$\hat{\boldsymbol{y}} = \boldsymbol{K}\boldsymbol{\alpha}, \tag{1.9}$$

with $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_{*,i}, \boldsymbol{x}_j)$.

1.6.2 Reproducing Kernel Hilbert Spaces

The general theory of Reproducing Kernel Hilbert Spaces (RKHS) was established in Aronszajn (1950) although its origins can be traced back to Szegő (1921) and Bergmann (1922).

Let $\mathcal{H}$ be a Hilbert space of real functions f defined on an index set $\mathcal{X}$. Then $\mathcal{H}$ is called a RKHS with an inner product $\langle \cdot \,|\, \cdot \rangle_{\mathcal{H}}$ and norm $\|f\|_{\mathcal{H}} = \sqrt{\langle f \,|\, f \rangle_{\mathcal{H}}}$ if there is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for every $\boldsymbol{x}$, $k(\boldsymbol{x}, \boldsymbol{z})$ as a function of $\boldsymbol{z}$ belongs to $\mathcal{H}$, and k complies with the reproducing property $\langle f(\cdot), k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}} = f(\boldsymbol{x})$ (Rasmussen & Williams, 2006).

The Moore-Aronszajn theorem in Aronszajn (1950) shows that to every RKHS $\mathcal{H}$ corresponds a unique positive definite function $k(\boldsymbol{x}, \boldsymbol{z})$ of two variables in $\mathcal{X}$ called the reproducing kernel of $\mathcal{H}$. In other words the RKHS uniquely determines k, and vice versa.

Interestingly, the function k behaves in $\mathcal{H}$ as the delta function does in L_2, and in particular $\langle k(\boldsymbol{x}, \cdot), k(\cdot, \boldsymbol{z})) \rangle_{\mathcal{H}} = k(\boldsymbol{x}, \boldsymbol{z})$. Now, for a set of functions $f(x) = \sum_{i=1}^{\infty} c_i \varphi_i(\boldsymbol{x})$ this Hilbert space is a RKHS as $\langle f(\boldsymbol{z}), k(\boldsymbol{z}, \boldsymbol{x}) \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{c_i \lambda_i \varphi_i(\boldsymbol{x})}{\lambda_i} = f(\boldsymbol{x})$ where the scalar product between two functions $f(\boldsymbol{x}) = \sum_{i=1}^{\infty} c_i \varphi_i(\boldsymbol{x})$ and $g(\boldsymbol{x}) = \sum_{i=1}^{\infty} d_i \varphi_i(\boldsymbol{x})$ is defined as $\langle f(\boldsymbol{x}), g(\boldsymbol{x}) \rangle_{\mathcal{H}} = \langle \sum_{i=1}^{\infty} c_i \varphi_i(\boldsymbol{x}), \sum_{i=1}^{\infty} d_i \varphi_i(\boldsymbol{x}) \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{c_i d_i}{\lambda_i}$ with kernel $k(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(\boldsymbol{x}) \varphi_i(\boldsymbol{z})$ where a sequence of positive numbers λ_i and linearly independent basis functions $\varphi_i(\boldsymbol{x})$ are assumed and the series converges (Suykens et al., 2002).

From Girosi (1998) the following problem is stated:

$$\min_{f\in\mathcal{H}} H\,|f| = \gamma\sum_{j=1}^{N} L(y_j - f(x_j)) + \frac{1}{2}\,\|f\|_k^2 \tag{1.10}$$

with $\mathcal{H}$ a RKHS with kernel k and L a loss function. This implies that functions in $\mathcal{H}$ have a unique expansion of the form $f(\boldsymbol{x}) = \sum_{i=1}^{\infty} c_i\varphi_i(\boldsymbol{x})$ with norm $\|f\|_k^2 = \sum_{i=1}^{\infty} c_i^2/\lambda_i$. Defininining $\alpha_j = \gamma L'(y_j - f(x_j))$ with L' the derivative of the loss function w.r.t. c_i we get $c_i = \lambda_i \sum_{j=1}^{N} \alpha_j \varphi_i(\boldsymbol{x}_j)$ and therefore

$$f(\boldsymbol{x}) = \sum_{i=1}^{\infty} c_i\varphi_i(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j). \tag{1.11}$$

Note that independently of the choice of L the solution to the problem is always a linear superposition of kernel functions.

The coefficients α_j are calculated as $\alpha_j = \gamma L'(y_j - \sum_{l=1}^{\infty} \alpha_l k(\boldsymbol{x}_j, \boldsymbol{x}_l))$ with $j = 1, \ldots, N$. When the least squares loss function is used this corresponds to the linear system

$$\boldsymbol{\alpha} = (\boldsymbol{\Omega} + \boldsymbol{I}/\lambda)^{-1}\boldsymbol{y}, \tag{1.12}$$

with $\boldsymbol{\Omega}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

1.6.3 Regularization Networks

Regularization networks were introduced in Poggio and Girosi (1990). The rationale behind this comes from the fact that almost all approximation schemes can be mapped into some kind of network that can be called also a neural network. In this context a network is a function composed by many basic functions.

The measure of the quality of an approximation can be defined by a distance function $g[f(\boldsymbol{X}), \hat{f}(\boldsymbol{W}, \boldsymbol{X})]$ with $f(\boldsymbol{X})$ the actual function and $\hat{f}(\boldsymbol{W}, \boldsymbol{X})$ the approximation. It is common to use norms as the distance functions (e.g. the L_2 norm).

Note that approximation functions $\hat{f}(\boldsymbol{W}, \boldsymbol{X}) : \mathbb{R}^n \to \mathbb{R}$ can be seen as corresponding to multilayer networks. For instance $\hat{f}(\boldsymbol{W}, \boldsymbol{X}) = \boldsymbol{w}^\top \boldsymbol{x}$ with $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^n$ corresponds to a network without hidden units. When a feature map is introduced, i.e. $\hat{f}(\boldsymbol{W}, \boldsymbol{X}) = \sum_{i=1}^{N} \boldsymbol{w}_i^\top \varphi_i(\boldsymbol{x})$, the approximation corresponds to a network with a layer of hidden units. It is known that this type of networks can approximate arbitrarily well any continuous multivariate function (Funahashi, 1989).

The combination of this network view with regularization is what gives birth to Regularization Networks. For a dataset $S = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^n \times \mathbb{R}\,|i = 1, \ldots, N\}$

that is meant to be approximated by a function f the regularization approach determines the function f through $\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i))^2 + \lambda \left\| Pf \right\|^2$ with P a constraint operator including the *a priori* information on the solution, $\|\cdot\|$ a norm function on the space to which Pf belongs and λ the regularization parameter. This leads to $f(\boldsymbol{x}) = \frac{1}{\lambda}\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i))G(\boldsymbol{x}, \boldsymbol{x}_i)$ implying that the solution of the regularization problem lies in an N-dimensional subspace of the space of smooth functions. A basis for this subspace is given by the N functions $G(\boldsymbol{x}, \boldsymbol{x}_i)$ (i.e. a Green's function). Let us now define $\alpha_i = (y_i - f(\boldsymbol{x}_i))/\lambda$ which leads to the expression

$$\boldsymbol{\alpha} = (\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}, \tag{1.13}$$

with $\boldsymbol{G}_{ij} = G(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Finally then, we get that

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i G(\boldsymbol{x}, \boldsymbol{x}_i). \tag{1.14}$$

In Wahba (1990) a similar result is derived through reproducing kernels.

1.6.4 Gaussian Processes

A way to understand Gaussian processes (GP) is as a generalization of Gaussian probability distributions. This means that an stochastic process governs the properties of functions in the same way as a probability distribution describes random variables (i.e. scalars or vectors). In loose terms a function can be seen as a very long vector where each entry in the vector specifies the function value $f(x)$ at a particular input x. This simplistic interpretation leads to an interesting concept: if the properties of the function at a finite number of points are wanted, inference in the Gaussian process will return the same answer if the infinitely many other points are ignored as if they had been taken into account. Furthermore, these answers are consistent with answers to any other finite queries for such function. In summary: A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen & Williams, 2006).

Another way to interpret GP models is that they are constructed from classical statistical models by replacing latent functions of parametric form by random processes with Gaussian prior (Seeger, 2004).

Let us have a training set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}$ with $i = 1, \ldots, N$ where $\boldsymbol{x} \in \mathbb{R}^n$ are inputs and $y \in \mathbb{R}$ are outputs. The matrix $\boldsymbol{X} \in \mathbb{R}^{N\times n}$ is simply the aggregation of the different $\boldsymbol{x}_i$, while the vector $\boldsymbol{y}$ is the equivalent for the outputs.

For a linear case, let us have $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{w}$ and $y = f(\boldsymbol{x}) + e$ with $\boldsymbol{x}$ the input vector, $\boldsymbol{w}$ a vector of weights of the linear model, f the function value, y the observed target value and $\boldsymbol{e}$ additive noise with an i.i.d. Gaussian distribution such that $\boldsymbol{x} \sim \mathcal{N}(0, \sigma_n^2)$. The model and the noise together give the likelihood $p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{X}\boldsymbol{w}, \sigma_n^2 \boldsymbol{I})$. For the prior a zero mean Gaussian can be used with covariance matrix $\boldsymbol{\Sigma}_p$ on the weights. Then $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_p)$. Using Bayes rule, the posterior can be calculated as

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \rightarrow p(\boldsymbol{w} \mid \boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w})}{p(\boldsymbol{y} \mid \boldsymbol{X})}. \tag{1.15}$$

The marginal likelihood is calculated as

$$p(\boldsymbol{y} \mid \boldsymbol{X}) = \int p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}) d\boldsymbol{w}. \tag{1.16}$$

The posterior is then in the form

$$p(\boldsymbol{w} \mid \boldsymbol{y}, \boldsymbol{X}) \sim \mathcal{N}(\bar{\boldsymbol{w}}, \boldsymbol{A}^{-1}), \tag{1.17}$$

with $\bar{\boldsymbol{w}} = \frac{1}{\sigma_n^2} \mathbf{A}^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ and $\boldsymbol{A} = \frac{1}{\sigma_n^2} \boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{\Sigma}_p^{-1}$.

Finally, for predictions in a test set an average over all possible parameter values, weighted by their posterior probability, is carried out. The predictive distribution is then given by

$$p(f_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(\frac{1}{\sigma_n^2} \boldsymbol{x}_*^\top \boldsymbol{A}^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \boldsymbol{x}_*^\top \boldsymbol{A}^{-1} \boldsymbol{x}_*). \tag{1.18}$$

To extend the previous approach to nonlinear cases a possibility is to first project the inputs into some high dimensional space through some sort of feature map and then apply the linear model in such space. Let $\varphi(\boldsymbol{x})$ be the function mapping the n-dimensional input to an n_h-dimensional (i.e. a higher dimensional space) feature space and $\boldsymbol{\Phi} \in \mathbb{R}^{n_h \times N}$ the aggregation of columns $\varphi(\boldsymbol{x})$ in the training set. The new model is then $f(\boldsymbol{x}) = \varphi(\boldsymbol{x})^\top \boldsymbol{w}$ where the parameters vector $\boldsymbol{w} \in \mathbb{R}^{n_h}$. The predictive distribution becomes

$$\begin{aligned} f_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y} \quad \sim \quad & \mathcal{N}(\boldsymbol{\varphi}_*^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{y}, \\ & \boldsymbol{\varphi}_*^\top \boldsymbol{\Sigma}_p \boldsymbol{\varphi}_* - \boldsymbol{\varphi}_*^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_p \boldsymbol{\varphi}_*), \end{aligned} \tag{1.19}$$

with $\varphi(\boldsymbol{x}_*) = \boldsymbol{\varphi}_*$, and $\boldsymbol{K} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}$. It can be seen that the feature map always appears in the form $\varphi(\boldsymbol{x})^\top \boldsymbol{\Sigma}_p \varphi(\boldsymbol{z})$ with $\boldsymbol{x}$ and $\boldsymbol{z}$ either in the training or test set. Let us define then

$$k(\boldsymbol{x}, \boldsymbol{z}) = \varphi(\boldsymbol{x}) \boldsymbol{\Sigma}_p \varphi(\boldsymbol{z}), \tag{1.20}$$

a covariance function or kernel which is an inner product w.r.t. $\boldsymbol{\Sigma}_p$. As $\boldsymbol{\Sigma}_p$ is positive definite $(\boldsymbol{\Sigma}_p^{1/2})^2 = \boldsymbol{\Sigma}_p$ and it is possible to define $\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{\Sigma}_p^{1/2} \varphi(\boldsymbol{x})$ so that an inner product representation is obtained as $k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{z})$. With this we can rewrite the predictive distribution for a new point $\boldsymbol{x}_*$

$$\begin{aligned} f_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y} \quad \sim \quad & \mathcal{N}(k(\boldsymbol{x}_*, \boldsymbol{x})(k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \\ & k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \boldsymbol{x})(k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})^{-1} k(\boldsymbol{x}, \boldsymbol{x}_*)). \end{aligned} \tag{1.21}$$

Finally, by defining

$$\boldsymbol{\alpha} = (k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \tag{1.22}$$

we can rewrite the above equation as

$$\begin{aligned} f_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y} \quad \sim \quad & \mathcal{N}(k(\boldsymbol{x}_*, \boldsymbol{x})\boldsymbol{\alpha}, \\ & k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \boldsymbol{x})(k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})^{-1} k(\boldsymbol{x}, \boldsymbol{x}_*)). \end{aligned} \tag{1.23}$$

There are several ways to interpret GP. The approach presented above corresponds to the so-called weight space view. Another (equivalent) approach is the function-space view (see Rasmussen and Williams (2006)).

1.6.5 Support Vector Machines

SVM usually is structured as follows: first the problem is formulated in the primal weight space as a constrained optimization problem, then the Lagrangian is formulated and the conditions for optimality are expressed. Finally the problem is solved in the dual space of the Lagrange multipliers (i.e. the support values) (Suykens et al., 2002).

Consider a training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ with inputs $\boldsymbol{x}_i \in \mathbb{R}^n$ and outputs $y_i \in \mathbb{R}$ with class labels $y_i \in \{+1, -1\}$ and the linear classifier $y_i = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x}_i + b)$. In the separable case $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$ for $i = 1, \ldots, N$, however, in the non-separable case it is necessary to tolerate missclassifications. In Cortes and Vapnik (1995) the extension to the nonseparable case was introduced. To do it slack variables are introduced as $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$ for $i = 1, \ldots, N$ and $\xi_i > 0$. Furthermore, this can be extended to the nonlinear case (V. N. Vapnik, 1998) through the use of functions $\varphi(\cdot)$ that map the input to a high dimensional space. Normally, these functions are known as feature maps. Once such mapping is done, the separating hyperplane can be constructed in the higher dimensional space.

The SVM theory can be extended to function estimation problems (V. N. Vapnik, 1995). To do this, the concept of Vapnik's ϵ-insensitive loss function is introdiced:

$$|y - f(\boldsymbol{x})| = \begin{cases} 0, & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon \\ |y - f(\boldsymbol{x})| - \epsilon, & \text{otherwise.} \end{cases} \tag{1.24}$$

The procedure is similar to that of the classification case. First the primal is formulated:

$$\begin{aligned}
&\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} J_P(\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\xi}^*) = \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + c\sum_{i=1}^{N}(\xi_i+\xi_i^*) \\
&\text{such that} \begin{cases} y_i - \boldsymbol{w}^\top\varphi(\boldsymbol{x}_i) - b \le \epsilon + \xi_i \text{ for } i = 1,\ldots,N \\ \boldsymbol{w}^\top\varphi(\boldsymbol{x}_i) + b - y_i \le \epsilon + \xi_i^* \text{ for } i = 1,\ldots,N \\ \xi_i, \xi_i^* \ge 0 \text{ for } i = 1,\ldots,N. \end{cases}
\end{aligned} \tag{1.25}$$

Then, the Lagrangian is stated and from the optimality conditions the dual problem is obtained:

$$\begin{aligned}
\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} J_D(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) &= -\tfrac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
&\quad -\epsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*) + \sum_{i=1}^{N}y_i(\alpha_i-\alpha_i^*) \\
\text{such that} &\quad \sum_{i=1}^{N}(\alpha_i-\alpha_i^*) = 0 \text{ and } \alpha_i,\alpha_i^* \in [0,c]
\end{aligned} \tag{1.26}$$

with c a positive real constant.

The dual representation of the model becomes then

$$\hat{y}_i(\boldsymbol{x}_*) = \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)k(\boldsymbol{x}_*,\boldsymbol{x}_i) + b, \tag{1.27}$$

where α_i and α_i^* are the solution to the quadratic programming (QP) problem and b follows from the complementarity Karush–Kuhn–Tucker (KKT) conditions.

1.7 LS-SVM

Support vector machines are introduced in Cortes and Vapnik (1995) to solve classification problems, but they can also be used for nonlinear function estimation or regression (V. Vapnik, 1998). Estimation of SVM typically requires the solution of a convex quadratic program due to the use of the ϵ-insensitive cost function and inequality constraints. In contrast to that, least-squares support vector machines (LS-SVM) (Suykens & Vandewalle, 1999; Suykens et al., 2002) use a formulation with a least-squares cost function and equality constraints. In that way, the involved computational complexity is reduced to that of solving a system of linear equations.

Besides classification and regression, SVM and LS-SVM have also been applied in the modeling of nonlinear dynamical systems (Lu, Sun, & Butts, 2017; Suykens,

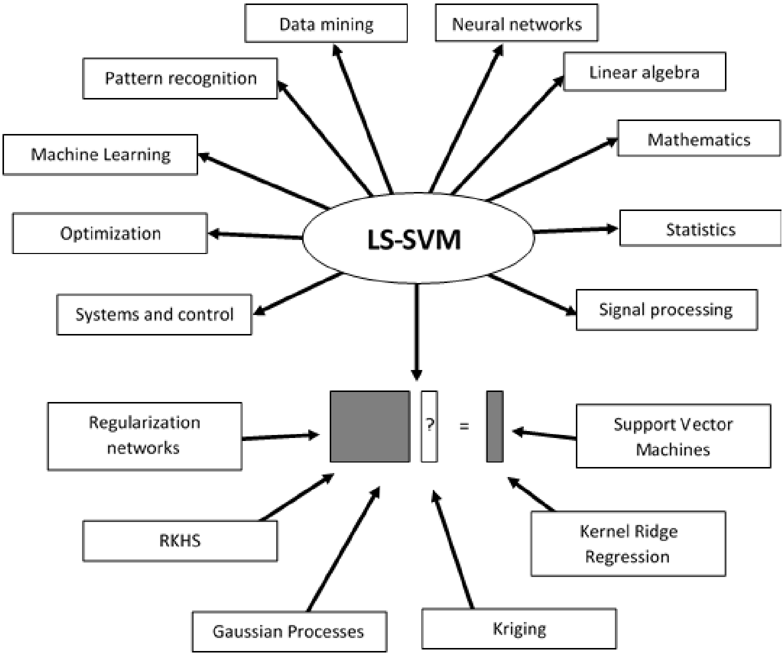


Figure 1.7: The interdisciplinarity of LS-SVM. LS-SVM is closely related to other kernel methods although its focus is mainly on primal-dual insights from optimization theory and neural networks (Suykens et al., 2002).

2001). The memory of the system is typically accounted for by adopting a NARX structure (Sjöberg et al., 1995), where the current output of the system is written as a nonlinear function of the current input and previous inputs and outputs. The estimation of the NARX model then boils down to a regression problem, for which SVM and LS-SVM are excellently suited. Taking into account feedback is less straightforward, but can be done using recurrent models (Suykens & Vandewalle, 2000). Besides this black-box NARX approach, also some approaches that incorporate structural information of the system can be found (see for example Goethals et al. (2005) for the identification of Hammerstein systems, Falck et al. (2012) for the identification of Wiener-Hammerstein systems in an LS-SVM context and Tötterman and Toivonen (2009) for the identification of Wiener systems in an SVM context).

The use of a least-squares cost function in LS-SVM has potential drawbacks. The least squares cost function in a linear regression context is optimal for white and/or Gaussian output noise, but not for correlated non-Gaussian noise. Moreover, in the absence of regularization, the method can be sensitive to outliers. Variants of the standard LS-SVM have been introduced to cope with these problems.

Weighted LS-SVM (Suykens, De Brabanter, Lukas, & Vandewalle, 2002) can deal with non-Gaussian errors by applying a weighted least-squares cost function, i.e. by associating individual weights to the error variables. The weights are chosen based on the results of an initial unweighted LS-SVM. This makes the method robust to outliers.

Instead of solving the problem at the level of the cost function, one can also work at the level of the kernel. RBF kernels are commonly used in LS-SVM models. An alternative to deal with correlated noise is to tune the kernel based on the noise that is present in the data. A bandwidth selection procedure based on bimodal kernels is proposed in De Brabanter, De Brabanter, Suykens, and De Moor (2011b).

In Laurain, Zheng, and Tóth (2011) and Laurain, Tóth, Piga, and Zheng (2015) an instrumental variable (IV) method in the LS-SVM framework is introduced. It extends LS-SVM to be consistent in general noise cases while maintaining its computational efficiency. The method is illustrated for affine NARX models in Laurain et al. (2011) and for general NARX and NARMAX (Chen & Billings, 1989; Sjöberg et al., 1995) models in Laurain et al. (2015).

In the framework of System Identification, LS-SVM has been applied before. Examples on well known benchmark data sets like the Wiener-Hammerstein data set (J. Schoukens et al., 2009) are available (e.g. De Brabanter et al. (2009) and Espinoza, Pelckmans, Hoegaerts, Suykens, and De Moor (2004)). Given the black box nature of LS-SVM, a natural improvement would be the ability to incorporate information about the structure of the system into the LS-SVM itself. This has been somewhat explored (e.g. Falck et al. (2012); Falck, Pelckmans, Suykens, and De Moor (2009)).

1.7.1 Function Estimation using Least Squares Support Vector Machines

Least Squares Support Vector Machines (LS-SVM) has been proposed within the framework of a primal-dual formulation (Suykens et al., 2002). Having a data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, the objective is to find a model

$$\hat{y} = \boldsymbol{w}^\top \varphi(\boldsymbol{x}) + b. \tag{1.28}$$

Here, $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ is the feature map to a high dimensional (possibly infinite) feature space, $\boldsymbol{w} \in \mathbb{R}^{n_h}$ is the weight vector, $\boldsymbol{x} \in \mathbb{R}^n$ is the input (for an input with n features), $\hat{y} \in \mathbb{R}$ represents the estimated output value, and b is the bias term.

A constrained optimization problem is then formulated:

$$\begin{aligned} \min_{\boldsymbol{w},b,e_i} \quad & \tfrac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \tfrac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ \text{subject to} \quad & y_i = \boldsymbol{w}^\top\varphi(\boldsymbol{x}_i) + b + e_i,\, i = 1, ..., N, \end{aligned} \tag{1.29}$$

with e_i the errors and γ the regularization parameter.

From the Lagrangian $\mathcal{L}(\boldsymbol{w}, b, e_i; \alpha_i) = \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N}\alpha_i(\boldsymbol{w}^\top\varphi(\boldsymbol{x}_i) + b + e_i - y_i)$, with $\alpha_i \in \mathbb{R}$ the Lagrange multipliers, the optimality conditions are derived:

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i) \\ \frac{\partial\mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^{N}\alpha_i = 0 \\ \frac{\partial\mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, i = 1, ..., N \\ \frac{\partial\mathcal{L}}{\partial\alpha_i} = 0 \rightarrow y_i = \boldsymbol{w}^\top\varphi(\boldsymbol{x}_i) + b + e_i, i = 1, ..., N. \end{cases} \tag{1.30}$$

Using Mercer's theorem Mercer (1909), the kernel matrix $\boldsymbol{\Omega}$ can be represented by the kernel function $\boldsymbol{\Omega}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top\varphi(\boldsymbol{x}_j)$ with $i, j = 1, ..., N$. It is important to note that in this representation $\varphi(\cdot)$ does not have to be explicitly known as it is implicitly used through the positive definite kernel function. A commonly used kernel is the radial basis function kernel (i.e. RBF kernel):

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{\sigma^2}\right), \tag{1.31}$$

where σ is the kernel parameter.

The dual formulation is obtained then from (1.30) by elimination of $\boldsymbol{w}$ and e_i:

$$\begin{bmatrix} 0 & \boldsymbol{1}_N^T \\ \boldsymbol{1}_N & \boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix} \tag{1.32}$$

with $\boldsymbol{y} = [y_1, ..., y_N]^\top$ and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^\top$. The resulting model is then:

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b. \tag{1.33}$$

1.7.2 NARX LS-SVM

The standard formulation of LS-SVM given in 1.7.1 can deal with static problems only in the sense that there are no recursive equations involved. The formulation however can be extended to problems including dynamics. To use the previous equations in the context of dynamical systems a combination with NARX models is required

$$\begin{aligned} \hat{y}(t) &= f(\boldsymbol{z}(t)), \text{ with} \\ \boldsymbol{z}(t) &= [y(t-1), y(t-2), \dots, y(t-r_a), \\ &\quad \boldsymbol{x}^\top(t), \boldsymbol{x}^\top(t-1), \dots, \boldsymbol{x}^\top(t-r_b)], \end{aligned} \tag{1.34}$$

where $\hat{y}(t) \in \mathbb{R}$ is the estimated output and $y(t-i)$, with $i = 1, \dots, r_a$ are the actual past outputs. $f(\cdot)$ is a smooth nonlinear mapping and $\boldsymbol{x}(t-j) \in \mathbb{R}^n$ with $j = 0, \dots, r_b$ are the current and past systems inputs.

With this structure a dynamical system with inputs $\boldsymbol{x}(t) \in \mathbb{R}^n$ and outputs $y(t) \in \mathbb{R}$ can be modeled with discrete time index t. The values of r_a and r_b represent the delays of outputs and inputs respectively.

Note that the training input for a system like this will consist of input data vectors $\boldsymbol{z} = [y(t-1), y(t-2), \dots, y(t-r_a), \boldsymbol{x}^\top(t), \boldsymbol{x}^\top(t-1), \dots, \boldsymbol{x}^\top(t-r_b)]_{t=r+1}^{r+N}$ with $r = \max(r_a, r_b)$. Also, the corresponding output values will be $\{y(t)\}_{t=r+1}^{r+N}$.

This formulation is inherently feed forward as the equation is not recursive. However it is a useful model for one-step-ahead predictions.

1.7.3 Thesis overview

This thesis is divided in four parts. The chapters in each part share similar methodologies or are based on similar concepts. A summary is presented below and on Fig. 1.8.

- Part I: In this part the best linear approximation method (BLA) (see Pintelon and Schoukens (2012) and Appendix A) is used in combination with LS-SVM for the identification of Hammerstein and Wiener systems. The chapters in this section are based on:

  ⋆ **Castro-Garcia, R.**, Tiels, K., Agudelo, O. M., Suykens, J. A. K. (2017). *Hammerstein System Identification through Best Linear Approximation Inversion and Regularization.* International Journal of Control. doi: 10.1080/00207179.2017.1329550. Available online at http://www.tandfonline.com/doi/abs/ 10.1080/00207179.2017.1329550.

  ⋆ **Castro-Garcia, R.**, Tiels, K., Schoukens, J., Suykens, J. A. K. (2015). *Incorporating Best Linear Approximation within LS-SVM-Based Hammerstein System Identification.* In proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015), Osaka, Japan. (pp. 7392 - 7397).

  - **Castro-Garcia, R.**, Suykens, J. A. (2016). *Wiener System Identification using Best Linear Approximation within the LS-SVM framework.* In proceedings of the 3rd Latin American Conference on Computational Intelligence. doi:10.1109/LA-CCI.2016.7885698.

- Part II: In this part system identification methods for Hammerstein and Wiener systems are offered where the common denominator is the use of the steady state time response of such systems. The chapters in this section are based on:

  - **Castro-Garcia, R.**, Agudelo, O. M., Tiels, K., Suykens, J. A. (2016). *Hammerstein system identification using LS-SVM and steady state time response*. In proceedings of the 15th European Control Conference (pp. 1063 – 1068).
  - **Castro-Garcia, R.**, Agudelo, O. M., Suykens, J. A. K. (2017c). *MIMO Hammerstein System Identification using LS-SVM and Steady State Time Response*. Accepted for publication in the proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI 2017). (Internal Report 17-23, ESAT-SISTA, KU Leuven. Leuven, Belgium).
  - Bottegal, G., **Castro-Garcia, R.**, Suykens, J. A. K. (2017a). *On the identification of Wiener systems with polynomial nonlinearity*. Accepted for publication in the proceedings of the 56th IEEE Conference on Decision and Control (CDC 2017). (Internal Report 17-55, ESAT-SISTA, KU Leuven. Leuven, Belgium).
  - Bottegal, G., **Castro-Garcia, R.**, Suykens, J. A. K. (2017b). *A two-experiment approach to Wiener system identification*. In Internal report 17-38, ESAT-SISTA, KU Leuven (Leuven, Belgium).

- Part III: In this part we take advantage of the Hammerstein structure to extract information about the dynamics of the system through the estimation of their impulse response. The chapters in this section are based on:

  - **Castro-Garcia, R.**, Agudelo, O. M., Suykens, J. A. K. (2017b). *Impulse Response Constrained LS-SVM modeling for Hammerstein System Identification*. In proceedings of the 20th world congress of the International Federation of Automatic Control (IFAC 2017), Toulouse, France. (pp. 14611 – 14616).
  - **Castro-Garcia, R.**, Agudelo, O. M., Suykens, J. A. K. (2017a). Impulse response constrained LS-SVM modeling for MIMO Hammerstein system identification. International Journal of Control. doi: 10.1080/00207179.2017.1373862. Available online at http://www. tandfonline.com/doi/abs/10.1080/ 00207179.2017.1373862.

- Part IV: In the final part of the thesis black box approaches are presented. The chapters in this section are based on:

Table 1.1: User guidelines for selecting one of the methods presented in this thesis.

		Transfer function model		Nonlinearity type		Inputs and outputs		User defined excitation signals	
	Chapter	Parametric	Non-parametric	Known	Unknown/ Difficult	SISO	MIMO	Yes	No
Hammerstein	2		x		x	x		x	
	3	x			x	x		x	
	5	x			x	x		x	
	6	x			x		x	x	
	8		x		x	x		x	
	9		x		x		x	x	
Wiener	4	x			x			x	
	7.1	x		x				x	
	7.2	x		x				x	
	7.3	x			x			x	
General	10				x	x	x		x

- ⋆ **Castro, R.**, Mehrkanoon, S., Marconato, A., Schoukens, J., Suykens, J. (2014). *SVD truncation schemes for fixed-size kernel models.* In proceedings of the International Joint Conference on Neural Networks. IJCNN 2014. Beijing, China, Jun. 2014 (pp. 3922-3929).
- ⋆ **Castro-Garcia, R.**, Tiels, K., Suykens, J. A. K. (2017). *Frequency Division LS-SVM for Nonlinear Modeling.* In Internal report 17-24, ESAT-SISTA, KU Leuven (Leuven, Belgium).

1.8 User guidelines

Many different options are introduced in this thesis for the identification of BONL systems. In this section, guidelines to choose the best methodology are offered. In Table 1.1 a summary is presented[1].

1.8.1 Hammerstein systems

Chapter 2

If the user has a nonparametric approximation to the transfer function of the underlying Hammerstein system (or can estimate one), this method can be used straightforwardly

[1] In Table 1.1, 7.1, 7.2 and 7.3 stand for the parametric, polynomial and non-parametric methods presented in chapter 7 respectively.

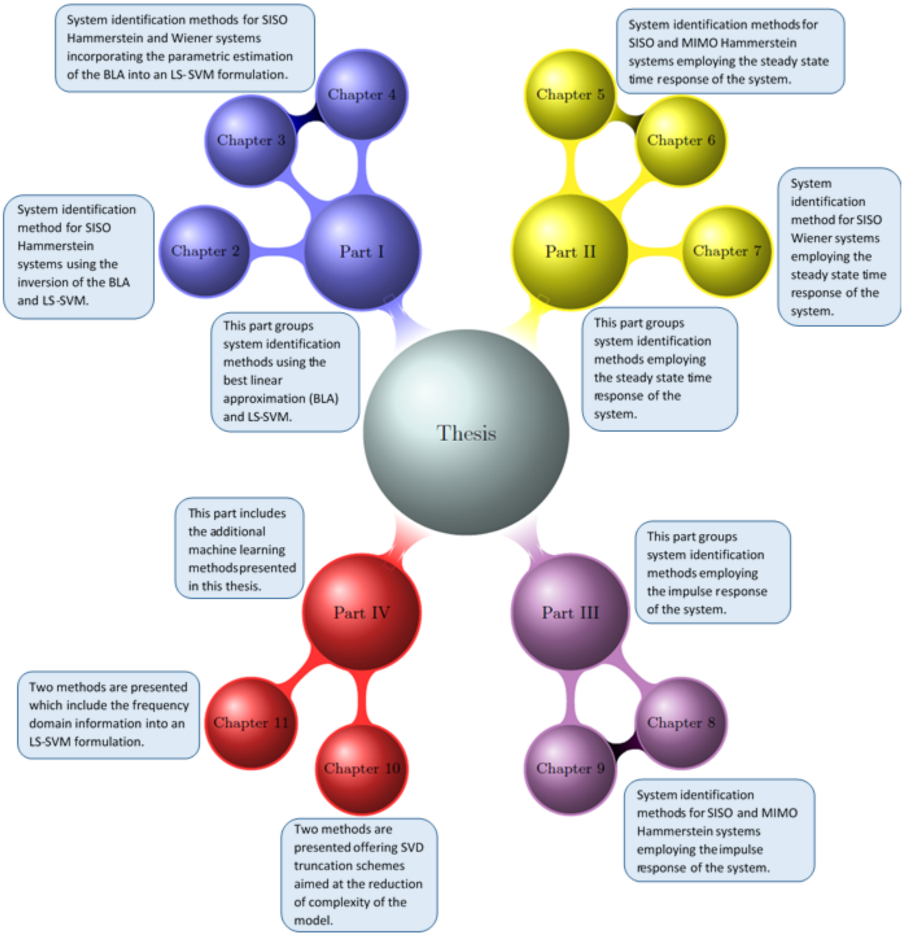

System identification methods for SISO Hammerstein and Wiener systems incorporating the parametric estimation of the BLA into an LS-SVM formulation.
Chapter 4
Chapter 3
System identification method for SISO Hammerstein systems using the inversion of the BLA and LS-SVM.
Chapter 2
Part I
This part groups system identification methods using the best linear approximation (BLA) and LS-SVM.
Chapter 5
System identification methods for SISO and MIMO Hammerstein systems employing the steady state time response of the system.
Chapter 6
Part II
Chapter 7
System identification method for SISO Wiener systems employing the steady state time response of the system.
This part groups system identification methods employing the steady state time response of the system.
Thesis
This part includes the additional machine learning methods presented in this thesis.
Part IV
Part III
This part groups system identification methods employing the impulse response of the system.
Two methods are presented which include the frequency domain information into an LS-SVM formulation.
Chapter 11
Chapter 10
Chapter 8
Chapter 9
System identification methods for SISO and MIMO Hammerstein systems employing the impulse response of the system.
Two methods are presented offering SVD truncation schemes aimed at the reduction of complexity of the model.

Figure 1.8: The structure of the thesis and a summarization of its parts and chapters.

and thus is a good option for Hammerstein system identification. In particular, the method excels when the class of problem are unknown or of a difficult-to-model nature thanks to the generalization properties of LS-SVM.

Chapter 3

This methodology is useful when a parametric approximation to the underlying transfer function is available or can be obtained. This is due to the fact that it shows how to easily incorporate that information into an LS-SVM formulation.

Chapters 5 and 6

These methodologies are very powerful as they allow a very good and straightforward modeling of the underlying nonlinearity. The user should be aware, however, that feeding the system with the required input signals can be difficult to achieve in practice due to their necessarily long times. For cases where this is not a problem or when a very good modeling of the underlying nonlinearity is required these methodologies constitute a very powerful tool for Hammerstein system identification.

Chapters 8 and 9

The type of input signals required for these methods are of a more common nature than the ones in Chapters 5 and 6 and can be generated in an easier way. Also, the flexibility of the methods w.r.t. the number of inputs and outputs and easiness of estimation make them the best of all the offered options in this thesis for Hammerstein system identification as long as the input signals can be chosen by the user.

1.8.2 Wiener systems

Chapter 4

Similarly to Chapter 3, this methodology is useful when a parametric approximation to the underlying transfer function is available or can be obtained. In such cases, the use of this method is straightforward.

Chapter 7 - Parametric and polynomial approaches

In the case that the user has a clear idea of the type of nonlinearities of the underlying Wiener system these methods can be very useful. The user should be aware that this knowledge is not always available and that the system has to be allowed to reach steady state and this can be a limitation.

Chapter 7 - Non-parametric approach

If the user is not aware of the type of nonlinearities of the underlying Wiener system, the non-parametric approach presented is a very attractive option as, thanks to the good generalization properties of LS-SVM, it can deal with different classes of problems. Once more, however, the system has to reach steady state and this can be a limitation.

1.8.3 General

Chapter 11

The methodologies presented in this chapter are especially appropriate for cases when the underlying structure of the system is unknown or when the user cannot define his own input signals. These methodologies show a better performance than the other powerful methodologies considered in this chapter when operating in similar conditions.

Part I

Best Linear Approximation and Least Squares Support Vector Machines

In this part, we explore the possibilities offered by using together the Best Linear Approximation (BLA) (Pintelon & Schoukens, 2012) and LS-SVM for system identification of block structured models. To this end, we consider Wiener and Hammerstein systems.

The methods are applied where they excel, that is, the BLA is used to model the linear parts and LS-SVM is used to model the nonlinear ones. As will be illustrated in the following chapters, this combination of methods can be done in several ways. The obtained results demonstrate that the proposed methods have a very good performance and therefore offer an attractive option.

Function Estimation using LS-SVM was presented in Section 1.7.1 while in Appendix A the BLA is presented completing the framework necessary for the methods introduced in the following chapters.

In Chapter 2 a method for Hammerstein system identification is presented. A novel approach for estimating the intermediate variable is presented allowing a clear separation of the identification steps. First, a nonparametric approximation to the linear block is obtained through the BLA of the system. Then an approximation to the intermediate variable is obtained using the inversion of the estimated linear block and the known output. Afterward a nonlinear model is calculated through LS-SVM using the estimated intermediate variable and the known input. To do this the regularization capabilities of LS-SVM play a crucial role. Finally, a parametric re-estimation of the linear block is made. This chapter is based on the work presented in Castro-Garcia, Tiels, Agudelo, and Suykens (2017).

Chapter 3 offers another methodology for Hammerstein system identification. First, a parametric approximation to the LTI block is obtained through the BLA. Then, the estimated coefficients of the transfer function from the LTI block are included in a modified LS-SVM formulation for modeling the system. This chapter is based on the work presented in Castro-Garcia, Tiels, Schoukens, and Suykens (2015).

Finally, in Chapter 4 a methodology for identifying Wiener systems is introduced based on the work presented in Castro-Garcia and Suykens (2016). Conceptually, this method is very similar to the one presented in Chapter 3: First, a parametric approximation to the LTI block is obtained through the BLA method and then the estimated coefficients of the transfer function from the LTI block are included in a modified LS-SVM formulation for modeling the system. However, given the difference in structures between Wiener and Hammerstein systems, the formulations and mathematical developments are substantially different.

Chapter 2

Hammerstein System Identification through Best Linear Approximation Inversion and Regularization

2.1 Introduction

In this chapter the Best Linear Approximation technique (BLA) (Pintelon & Schoukens, 2012) is used in order to find an initial estimate of the linear block. With this model the identification of the nonlinear model can proceed. It is important to use a technique that includes regularization as will be shown later. For this task we use Least Squares Support Vector Machines (LS-SVM) (Suykens et al., 2002) given its well known generalization properties and its incorporated regularization mechanisms. In order to link these two steps a novel method to obtain an estimation of the intermediate variable (i.e. $x(t)$ in Fig. 1.1) is propsed. A mix of techniques is used then and they are applied where they excel, that is, we use the BLA to model the linear part and LS-SVM to model the nonlinear one.

This chapter is based on:
Castro-Garcia, R., Tiels, K., Agudelo, O. M., Suykens, J. A. K. (2017). *Hammerstein System Identification through Best Linear Approximation Inversion and Regularization.* International Journal of Control. doi: 10.1080/00207179.2017.1329550. Available online at http://www.tandfonline.com/doi/abs/10.1080/00207179.2017.1329550.

The proposed methodology can be separated in four stages:

- The system's BLA is calculated and used as an approximation to the linear block.
- The intermediate variable is estimated using the inversion of the approximated linear block and the known output.
- An LS-SVM model is trained using the known input and the estimated intermediate variable.
- On an independent data set, the linear block is re-estimated using the result from applying the input to the estimated model of the nonlinear block (i.e. the newly estimated intermediate variable) and the known output.

The proposed method uses a backwards approach as defined in Sun, Liu, and Sano (1999) where the linear dynamic part is identified first, then the intermediate variable is estimated and finally the nonlinear part is modeled. This type of approach is also used in other works like Bai and Fu (2002), Bai (1998) and J. Wang et al. (2009). Note however that our approach differs from the Blind Approach defined in Bai and Fu (2002) as the linear block is estimated using both input and output signals. Additionally, in this work we make a final refinement of the estimation of the linear block after having a model for the nonlinear block.

The idea of using the inversion of the estimated linear block is not new but is in general regarded as a bad idea. As explained in Crama and Schoukens (2001b), among other problems, the inversion of the found model implies that at those frequencies where the amplitude of the transfer function is small, the noise will be amplified. Nevertheless, the concept has been employed for example in Bai and Fu (2002) and Bai (2004) where the effect of noise is not explored deeply. In contrast, in the present work the output of the Hammerstein system is measured in the presence of high levels of white Gaussian additive noise $n_y(t)$ (see Fig. 1.1). Additionally, some common assumptions like a known order of the linear system are not necessary here.

In addition to the problem posed by the noise, the possibility of the system being non-minimum phase is a serious concern to this type of approach. In the presented method, we offer a way to work around these problems.

This chapter is organized as follows: In Section 2.2 the proposed method is presented. Section 2.3 illustrates the results found when applying the described methodology on three simulation examples, one of which is based on a real life application. Finally, in Section 2.4, the conclusions are presented.

2.2 Proposed Method

2.2.1 Inversion in the frequency domain

When dealing with system identification of Hammerstein systems it would be desirable to be able to use $y(t)$, the output of the system, and the inverse of the estimated linear block to obtain an approximation to the intermediate variable i.e. $\hat{x}(t)$. However, it is often the case that this inversion is not straightforward. For instance, if the identified linear system is parametric and has zeros outside the unit circle (i.e. for the discrete time case), when the system is inverted, those zeros would become poles and the resulting system would be unstable, thus making the operation unfeasible.

Once an approximation to the linear block is obtained through the BLA (i.e. $G_{BLA}(k)$, see Appendix A), it is possible to use it to obtain an approximation to the intermediate variable $\hat{X}(k)$. To do this, first $G_{BLA}(k)$ is inverted at each frequency:

$$G_{BLA}^{-1}(k) = \frac{1}{|G_{BLA}(k)|} \exp\left(-\Theta_{G_{BLA}(k)} j\right) \tag{2.1}$$

with $|G_{BLA}(k)|$ the magnitude and $\Theta_{G_{BLA}(k)}$ the phase of $G_{BLA}(k)$ at each frequency. From this representation it is clear that the proposed method will be able to invert linear blocks even if they would result in unstable systems if inverted in the time domain. This nice property comes from the fact that this inversion is done in the frequency domain.

In a Bode plot, this would look as the example presented in Fig. 2.1 where the product of the magnitudes is one and the sum of the phases is zero. The shown system corresponds to

$$G_0(q) = \frac{q^6 + 0.8q^5 + 0.3q^4 + 0.4q^3}{q^6 - 2.789q^5 + 4.591q^4 - 5.229q^3 + 4.392q^2 - 2.553q + 0.8679}. \tag{2.2}$$

Note that the sudden perturbation from 33% of the sampling frequency is nothing more than an artifact due to the chosen excitation signal employed for computing the BLA.

Once $G_{BLA}^{-1}(k)$ is obtained, and having $Y(k)$ (i.e. the representation of the known $y(t)$ in the frequency domain), their product will result in $\hat{X}(k)$:

$$\hat{X}(k) = \frac{|Y(k)|}{|G_{BLA}(k)|} \exp\left(\left(\Theta_{Y(k)} - \Theta_{G_{BLA}(k)}\right) j\right) \tag{2.3}$$

with $|Y(k)|$ the magnitude and $\Theta_{Y(k)}$ the phase of $Y(k)$ at each frequency k.

From $\hat{X}(k)$, its corresponding time domain representation $\hat{x}(t)$ can be recovered and with it, the identification of the nonlinear block, through LS-SVM in this case, can

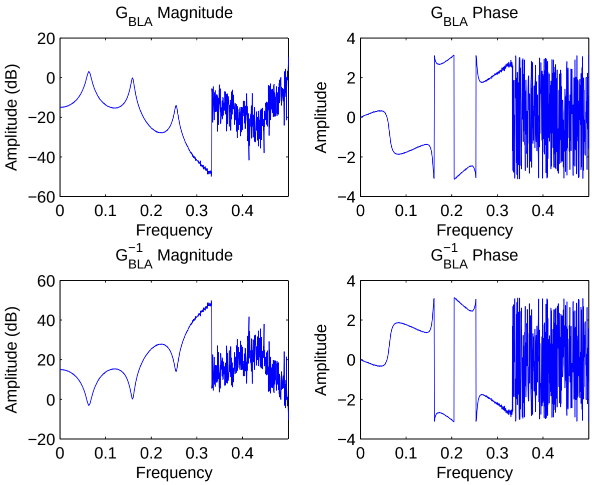


Figure 2.1: Example: Bode plot for $G_{BLA}(k)$ (upper left and right) and its corresponding inverted result $G_{BLA}^{-1}(k)$ (lower left and right).

proceed. Note that this is possible due to the fact that the input to the nonlinear block $u(t)$ is known and an approximation to its output $\hat{x}(t)$ is now available.

It is evident from the multiplication in (2.3) that whatever noise is contained in $Y(k)$ will be propagated into this intermediate variable estimation. Normally this would be a problem and obviously is an undesired effect. In fact, $\hat{x}(t)$ will be in general a poor approximation to the actual $x(t)$ since beside the backpropagated noise it is only estimated in the frequency band where $G_{BLA}(k)$ was obtained. However, it is of paramount importance to highlight the fact that $\hat{x}(t)$ is used here exclusively to train the LS-SVM model and that due to the regularization properties of LS-SVM, the aforementioned issues can be overcome (i.e. see Section 2.2.2 and Figure 2.4).

2.2.2 Role of regularization

To illustrate the effect of the regularization, it will be shown how the changes in the noise level affect the resulting model.

Given that $\hat{x}_T(t)$ (i.e. the estimated intermediate variable of the training data) is defined as a multiplication in the frequency domain, it can be seen naturally as a convolution in the time domain. To express this, we will define $\boldsymbol{M}_{\boldsymbol{G}_{BLA}^{-1}}$ as a Toeplitz matrix containing the time domain representation of $G^{-1}_{BLA}(k)$:

$$\hat{\boldsymbol{x}}_{\boldsymbol{T}} = \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{y}. \tag{2.4}$$

Note that here $\boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \in \mathbb{R}^{N \times N}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{T}},\ \boldsymbol{y} \in \mathbb{R}^{N}$ with $\boldsymbol{y}$ the measured output and $\hat{\boldsymbol{x}}_{\boldsymbol{T}}$ the estimated intermediate variable of the training data (i.e. it comes from $Y(k)$ and the inversion of $G_{BLA}(k)$).

Here, $\hat{\boldsymbol{x}}_{\boldsymbol{T}}$ is used in (1.32) instead of $\boldsymbol{y}$, therefore, from the LS-SVM formulation, we have:

$$\hat{\boldsymbol{x}}_{\boldsymbol{T}} = \left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)\boldsymbol{\alpha} + \mathbf{1}_{\boldsymbol{N}} b, \tag{2.5}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{\Omega}\boldsymbol{\alpha} + \mathbf{1}_{\boldsymbol{N}} b. \tag{2.6}$$

Note that $\tilde{\boldsymbol{x}}$ corresponds to the predicted values of the intermediate variable through the evaluation of the found model on the training data.

From (2.4) and (2.5) into (2.6), we can rewrite $\tilde{\boldsymbol{x}}$ as:

$$\begin{aligned} \tilde{\boldsymbol{x}} &= \hat{\boldsymbol{x}}_{\boldsymbol{T}} - \frac{\boldsymbol{\alpha}}{\gamma} \\ &= \hat{\boldsymbol{x}}_{\boldsymbol{T}} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1}(\hat{\boldsymbol{x}}_{\boldsymbol{T}} - \mathbf{1}_{\boldsymbol{N}} b) \\ &= \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{y} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1}(\boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{y} - \mathbf{1}_{\boldsymbol{N}} b). \end{aligned} \tag{2.7}$$

Now, if we want to see how $\tilde{\boldsymbol{x}}$ changes if the measurement noise $\boldsymbol{n}_{\boldsymbol{y}}$ changes, we have:

$$\begin{aligned} \tilde{\boldsymbol{x}} + \boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}} =& \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}}(\boldsymbol{y} + \boldsymbol{\Delta}_{\boldsymbol{n}_{\boldsymbol{y}}}) - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1}(\boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}}(\boldsymbol{y} + \boldsymbol{\Delta}_{\boldsymbol{n}_{\boldsymbol{y}}}) - \mathbf{1}_{\boldsymbol{N}} b) \\ =& \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{y} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1}(\boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{y} - \mathbf{1}_{\boldsymbol{N}} b) + \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{\Delta}_{\boldsymbol{n}_{\boldsymbol{y}}} \\ & - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1} \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{\Delta}_{\boldsymbol{n}_{\boldsymbol{y}}} \\ =& \tilde{\boldsymbol{x}} + \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{\Delta}_{\boldsymbol{n}_{\boldsymbol{y}}} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1} \boldsymbol{M}_{\boldsymbol{G}^{-1}_{\boldsymbol{BLA}}} \boldsymbol{\Delta}_{\boldsymbol{n}_{\boldsymbol{y}}}. \end{aligned} \tag{2.8}$$

From (2.8) we get then:

$$\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}} = \left(\boldsymbol{I} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1}\right) \boldsymbol{M}_{\boldsymbol{G}_{BLA}^{-1}} \boldsymbol{\Delta}_{\boldsymbol{n_y}}. \tag{2.9}$$

Now, from (2.9) let us define:

$$\boldsymbol{W} = \boldsymbol{I} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1}. \tag{2.10}$$

Hence

$$\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}} = \boldsymbol{W}\boldsymbol{M}_{\boldsymbol{G}_{BLA}^{-1}} \boldsymbol{\Delta}_{\boldsymbol{n_y}}. \tag{2.11}$$

Here $\boldsymbol{W}$ is the matrix determining the effect of the change of the measurement noise in the model. Note also that it can be rewritten as

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{I} - (\gamma\boldsymbol{\Omega} + \boldsymbol{I})^{-1} \\ &= \boldsymbol{\Omega}\left(\boldsymbol{\Omega} + \tfrac{\boldsymbol{I}}{\gamma}\right)^{-1}. \end{aligned} \tag{2.12}$$

Theorem 1. *In* (2.11) $\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}}$ *is upper bounded as follows* $\|\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}}\|_2 \leq \left\|\boldsymbol{M}_{\boldsymbol{G}_{BLA}^{-1}} \boldsymbol{\Delta}_{\boldsymbol{n_y}}\right\|_2$

Proof. Let λ_i be an eigenvalue of $\boldsymbol{\Omega}$ which is a symmetric positive semi-definite matrix. The eigenvalues of $\gamma\boldsymbol{\Omega} + \boldsymbol{I}$ are equal to $\gamma\lambda_i + 1$.

As $\gamma\boldsymbol{\Omega} + \boldsymbol{I}$ is a square non-singular matrix, the eigenvalues of the matrix $(\gamma\boldsymbol{\Omega} + \boldsymbol{I})^{-1}$ are $\frac{1}{(\gamma\lambda_i+1)}$ and the eigenvalues of $\boldsymbol{W} = \boldsymbol{I} - (\gamma\boldsymbol{\Omega} + \boldsymbol{I})^{-1}$ are then $\left(1 - \frac{1}{(\gamma\lambda_i+1)}\right) = \frac{\gamma\lambda_i}{(\gamma\lambda_i+1)} = \frac{\lambda_i}{(\lambda_i+\frac{1}{\gamma})}$.

Given that $\boldsymbol{W}$ is symmetric, $\left(\frac{\lambda_i}{(\lambda_i+1)}\right)^2$ is an eigenvalue of $\boldsymbol{W}^\top\boldsymbol{W}$.

Note that $0 \leq \frac{\lambda_i}{(\lambda_i+\frac{1}{\gamma})} \leq 1$ as by definition $\gamma > 0$ and $\lambda_i \geq 0$. Therefore $0 \leq \left(\frac{\lambda_i}{(\lambda_i+\frac{1}{\gamma})}\right)^2 \leq 1$.

Finally, note that $\|\boldsymbol{W}\|_2 = \sqrt{\max\limits_i \left(\frac{\lambda_i}{(\lambda_i+\frac{1}{\gamma})}\right)^2}$.

From (2.11) we can write the following inequality: $\|\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}}\|_2 \leq \|\boldsymbol{W}\|_2 \left\|\boldsymbol{M}_{\boldsymbol{G}_{BLA}^{-1}} \boldsymbol{\Delta}_{\boldsymbol{n_y}}\right\|_2 \leq \left\|\boldsymbol{M}_{\boldsymbol{G}_{BLA}^{-1}} \boldsymbol{\Delta}_{\boldsymbol{n_y}}\right\|_2$ as $0 \leq \|\boldsymbol{W}\|_2 \leq 1$ □

Note that given the properties of $\boldsymbol{W}$ and the expression in (2.11) we can expect that the effect of $\boldsymbol{\Delta}_{\boldsymbol{n_y}}$ will be generally dampened by $\boldsymbol{W}$. This is very relevant as $\boldsymbol{W}$ is heavily dependent on the regularization term γ. See Fig. 2.6 in Section 2.3.2 for an example of the behavior of $||\boldsymbol{W}||_2$ and the effect of γ on it.

Let us consider the case where $\gamma \to 0$ while bearing in mind that the values of the kernel matrix $\boldsymbol{\Omega}_{i,j} \in [0,1] \, \forall i,j$ when using the Gaussian RBF kernel, where i and j denote the row and column evaluated. Note that in this case, the weight given to the errors between the output of the training data set and the points obtained with the model is very small (see (1.32)).

$$\begin{aligned} \boldsymbol{W} =& \boldsymbol{I} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1} \\ =& \boldsymbol{I} - \frac{1}{\gamma}\left(\frac{\gamma\boldsymbol{\Omega} + \boldsymbol{I}}{\gamma}\right)^{-1} \\ \approx& \boldsymbol{I} - \frac{1}{\gamma}\left(\frac{\boldsymbol{I}}{\gamma}\right)^{-1} = \boldsymbol{0}. \end{aligned} \tag{2.13}$$

Then:

$$\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}} \approx \boldsymbol{0}. \tag{2.14}$$

Consider now the case where $\gamma \to \infty$. The errors between the training points and the points obtained with the model are extremely important (again, see (1.29)). In other words, the estimated model would try to follow the output of the training data set as well as possible.

$$\boldsymbol{W} = \boldsymbol{I} - \frac{1}{\gamma}\left(\boldsymbol{\Omega} + \frac{\boldsymbol{I}}{\gamma}\right)^{-1} \approx \boldsymbol{I} - \frac{\boldsymbol{\Omega}^{-1}}{\gamma} \approx \boldsymbol{I}. \tag{2.15}$$

Then:

$$\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}} \approx \boldsymbol{M}_{\boldsymbol{G}_{\boldsymbol{BLA}}^{-1}} \boldsymbol{\Delta}_{\boldsymbol{n_y}}. \tag{2.16}$$

This result was to be expected since the model will try to follow the output of the training data set. Any change in the training points will result in a direct change in the behavior of the model as there is no regularization at all.

It is important to remember that the Toeplitz matrix $\boldsymbol{M}_{\boldsymbol{G}_{\boldsymbol{BLA}}^{-1}}$ is used such that a convolution in the time domain can be represented. If the expression in (2.16) is taken to the frequency domain, it simply becomes a multiplication:

$$\Delta_{\tilde{X}}(k) \approx G_{BLA}^{-1}(k)\Delta_{N_Y}(k). \tag{2.17}$$

From (2.17) it is evident that the way $\Delta_{N_Y}(k)$ is going to affect $\Delta_{\tilde{X}}(k)$, or equivalently how $\boldsymbol{\Delta_{n_y}}$ affects $\boldsymbol{\Delta_{\tilde{x}}}$, depends directly on the frequency power distributions of $G_{BLA}^{-1}(k)$ and $\Delta_{N_Y}(k)$ as there would be no regularization in use. This means that in the worst case scenario, most of the power of $\Delta_{N_Y}(k)$ would be in the passband of $G_{BLA}^{-1}(k)$. In this case, the effect of $\Delta_{N_Y}(k)$ would be amplified. It is possible, on the other hand, that most of the power of $\Delta_{N_Y}(k)$ is out of the passband of $G_{BLA}^{-1}(k)$. In this case, the effect of $\Delta_{N_Y}(k)$ would be dampened.

2.2.3 Method Summary

The proposed method finally consists of four parts. First, the BLA of the system (i.e. a non-parametric $G_{BLA}(k)$) is calculated. Second, using $G_{BLA}(k)^{-1}$ and the frequency domain representation of $y(t)$ (i.e. $Y(k)$) an approximation to the intermediate variable $\hat{X}_T(k)$ is obtained and from it, its corresponding time domain representation $\hat{x}_T(t)$. Next, an LS-SVM model is trained using $u(t)$ as input and $\hat{x}_T(t)$ as output. Finally, the linear block is re-estimated using a newly calculated intermediate variable (i.e. the result from applying a new input to the estimated nonlinear block) and the corresponding known output. A summary of the method presenting the main steps is shown in Algorithm 1. Note that for each of the data sets mentioned, several realizations are used (see Section 2.3.2).

It is important to highlight that in Algorithm 1, steps 1 to 6 correspond to the System Identification part, while steps 7 to 9 correspond to the evaluation part.

2.3 Experimental Results

In this section a practical illustration of the proposed method is presented through synthetic examples. First, the specific characteristics of the employed signals will be described. Then a didactical example will be offered where the steps of the proposed method described in Algorithm 1 are shown. Afterward, the effects of the noise on the proposed method is illustrated in the previous example and in a new example which includes a hard to model nonlinearity. Finally, we offer a third example based on a real life application and provide a performance comparison with different methods for the three examples.

Note that the particulars (e.g. amplitudes, frequency bands, sampling frequencies, etc.) of the signals used were picked for the corresponding examples and are not intended as general recommendations for the proposed method. Also, in order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D.

Algorithm 1 Hammerstein System Identification through BLA inversion and LS-SVM techniques.

```
Input: Random phase multisine signal $u_1(t)$ and its corresponding output $y_1(t)$; Ramp signal $u_2(t)$ and its corresponding output $y_2(t)$. Multilevel Pseudo Random Signal $u_3(t)$ and its corresponding output $y_3(t)$. Multilevel Pseudo Random Signal $u_{test}(t)$.
Output: Evaluation of the test output signal $y_{test}(t)$;
Eliminate the first data points in $u_1(t)$ and $y_1(t)$ to reduce the effect of the transients.
Get the frequency representation of the transient free signals (i.e. $U_1(k)$ and $Y_1(k)$).
Obtain $G_{BLA}(k)$ and $G_{BLA}^{-1}(k)$ from several realizations of $U_1(k)$ and $Y_1(k)$.
Get an approximation to the intermediate variable in the frequency domain: $\hat{X}_T(k) = G_{BLA}^{-1}(k)Y_2(k)$.
Get $\hat{x}_T(t)$, the time domain representation of $\hat{X}_T(k)$, and together with $u_2(t)$ train an LS-SVM model to represent the nonlinearity;
Apply $u_3(t)$ to the estimated nonlinear model. Using the resulting $\tilde{x}_3(t)$ and the known output $y_3(t)$, estimate the linear block $G_{final}(k)$;
Apply $u_{test}(t)$ to the estimated nonlinear model. From the resulting $\tilde{x}_{test}(t)$ obtain its frequency representation. Apply $\tilde{X}_{test}(k)$ to $G_{final}(k)$ to obtain the estimation of the output in the frequency domain $\hat{Y}_{test}(k)$.
Finally, from $\hat{Y}_{test}(k)$ obtain the corresponding time domain representation $\hat{y}_{test}(t)$;
return $\hat{y}_{test}(t)$;
```

2.3.1 Signals description

In this section, the signals mentioned in Algorithm 1 are further explained.

Signal $u_1(t) = \sum_{k=1}^{F} |U_k| \cos(2\pi k \frac{f_s}{N} t + \phi_k)$ is a full random-phase multisine with a random harmonic grid in the excited frequency range $\{0, 10000\}$Hz. Note that here, F stands for the excited frequencies, N is the number of samples and $|U_k|$ denotes the amplitude used at each frequency. The sampling frequency f_s is equal to 78125Hz. In this case, all $|U_k| \neq 0$ are chosen equal to each other such that $u_1(t)$ has an rms value of 0.3. The phases ϕ_k are uniformly and randomly distributed between 0 and 2π. A ramp signal with 45 degrees slope was used to generate $u_2(t)$. Finally, Multilevel Pseudo Random Signals with 2% switching probability and amplitude values drawn from a uniform distribution were used to generate $u_3(t)$ and $u_{test}(t)$.

A summary and further details describing the used signals can be found in Table 2.1.

Name	**Points used**	**Max. Value**	**Min. Value**	**Description**
$u_1(t)$	5000	-	-	Random Phase Multisine Signal.
$u_2(t)$	1000	15	-15	Ramp signal with 45 degrees slope.
$u_3(t)$	1000	10	-10	Multilevel Pseudo Random Signal.
$u_{test}(t)$	1000	10	-10	Multilevel Pseudo Random Signal.

Table 2.1: Summary of input signals used.

It is important to note that even though the signals have the number of points noted in Table 2.1, it is possible to perform an interpolation of such signals to have more points.

2.3.2 Method steps

In this section the proposed methodology was applied to one system in the discrete time domain. This system is of didactical nature and is used to illustrate the method steps. It was generated through a nonlinear block:

$$x(t) = u(t)^3 \text{ with } u(t) \in [-15, 15], \tag{2.18}$$

and a linear block:

$$y(t) = \frac{B_1(q)}{A_1(q)} x(t) \tag{2.19}$$

where

$$B_1(q) = 0.008935q^3 - 0.004525q^2 - 0.004525q + 0.008935 \tag{2.20}$$

and

$$A_1(q) = q^3 - 2.564q^2 + 2.218q - 0.6456. \tag{2.21}$$

The linear and nonlinear blocks in this example (later on referred to as Example 1) are depicted in Fig. 2.2. The LTI block was chosen with a sharp zero in order to illustrate how this affects the method.

White Gaussian noise with zero mean was applied to the output of the system with a SNR of 10 dB. For each of the stages in the procedure, 2 periods and 5 realizations of the signals were used.

First, a non-parametric $G_{BLA}(k)$ was estimated as shown in Appendix A through the use of a random phase multisine input and its corresponding output (see Fig. 2.3 for the results of this step in the example). Afterward $G_{BLA}^{-1}(k)$ was calculated as explained

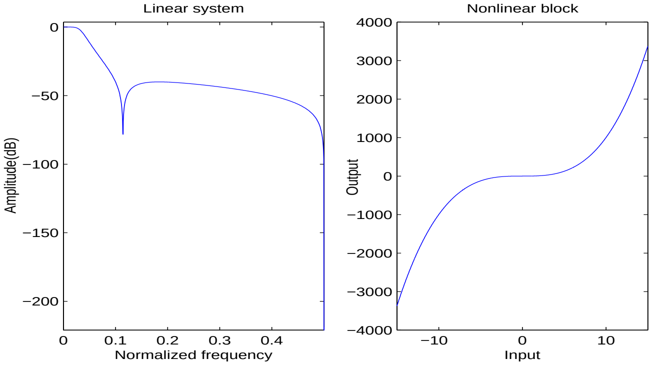


Figure 2.2: (Left) Linear system in normalized frequency and dB. (Right) Nonlinear block.

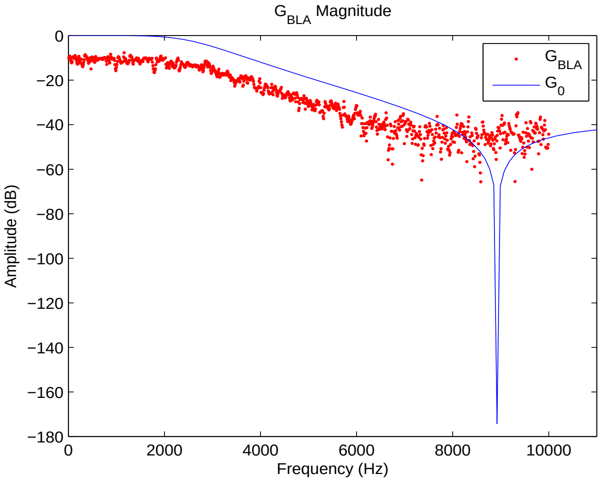


Figure 2.3: Nonparametric BLA.

in Section 2.2. Next, the estimation of the intermediate variable $\hat{x}_T(t)$ was carried out. Then, the nonlinear block was estimated through the use of LS-SVM using $u_2(t)$ as input and $\hat{x}_T(t)$ as output. Note that this $u_2(t)$ still corresponds to the same signal used to estimate $\hat{x}_T(t)$. Fig. 2.4 displays the results of this step.

Note that there is a rescaling of the approximated nonlinear block before overlapping

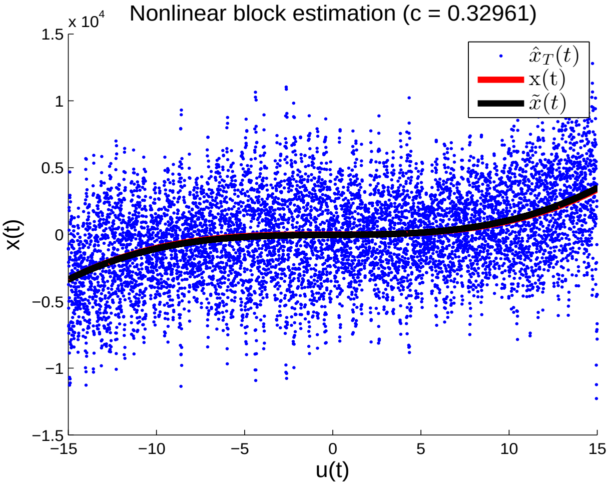


Figure 2.4: Nonlinear block estimation. Here, c corresponds to a rescaling factor. An interpolation to increase the number of points was performed and thus the extended number of points.

it to the actual one. The actual difference in scaling has no effect on the input-output behavior of the Hammerstein system (i.e. any pair of $\{f(u(t))/\eta,\ \eta G(q)\}$ with $\eta \neq 0$ would yield identical input and output measurements). Other than this rescaling, it is evident that the reproduction of the nonlinear block is quite accurate. In fact, in Fig. 2.5, the comparison between the actual $x_{test}(t)$ and the estimated $\tilde{x}_{test}(t)$ is shown. This comparison is carried out for illustrative purposes and does not have any role at all in the procedure. As can be seen, the reconstruction is accurate up to a scaling factor.

In Fig. 2.6, the behavior of $\|\boldsymbol{W}\|_2$ is displayed for different values of σ and γ. As can be seen, $\|\boldsymbol{W}\|_2$ is bounded between 0 and 1 as explained in Theorem 1.

Also, in order to show the effect of parameter tuning during the modeling of the nonlinear block, Figures 2.7 and 2.8 are presented. In these figures, alternatively one of the parameters (i.e. σ or γ) is fixed while different values for the other are tried at the training and test set and the resulting $\%$MAE is presented (i.e. see Appendix D). The fixed values correspond to the selected parameters through Coupled Simulated Annealing (Xavier-de Souza, Suykens, Vandewalle, & Bollé, 2009) followed by a Simplex approach for fine tuning under a 10-fold crossvalidation scheme (i.e. see LS-SVMlab v1.8).

As can be seen, neither of the parameters is selected at the corresponding lowest error in the training set, however, in the test set these parameters prove to be very effective.

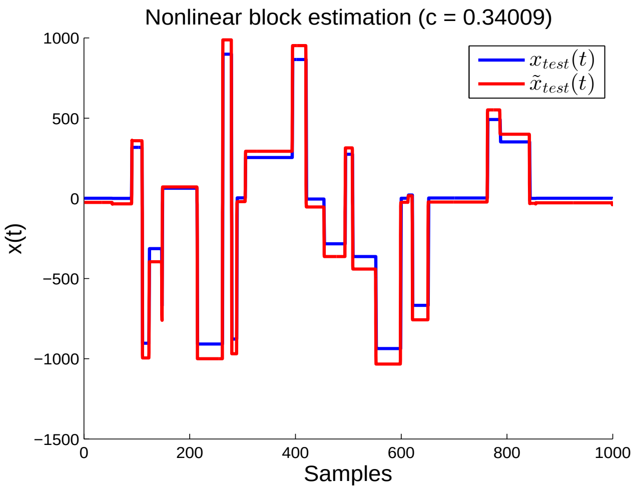


Figure 2.5: Comparison between the actual $x_{test}(t)$ and the estimated $\hat{x}_{test}(t)$. Here, c corresponds to a rescaling factor.

Note that even though it is not possible to guarantee the absolute minimum error in the test set, a very good result is obtained.

Having performed the previous steps, the last remaining thing to do is to re-estimate the linear block. To do this, a Least Squares (LS) approach is used. The result can be seen in Fig. 2.9, and in Fig. 2.10 the final estimation for the test set is shown.

2.3.3 Noise effect analysis

For evaluating how the noise affects the performance of the method, 100 Monte Carlo simulations were carried out in a test set for each of four different levels of SNR.

The results of the 100 Monte Carlo simulations are summarized in Fig. 2.11. As can be seen, the performance of the model is consistent independently of the level of noise (i.e. the medians of the different simulations oscillate in a small range between 2.8101% and 3.4928%). This phenomenon can be explained by the particular shape of the linear block in this system. As can be seen in Figs. 2.3 and 2.9, the zero at 8926Hz is not modeled properly due to the high level of noise present. In Fig. 2.12, the same modeling is shown for different SNR levels. It is clear that as the level of noise decreases, the modeling of the zero improves. This better modeling has a drawback for the proposed methodology: When the model is inverted, a pole will appear at the same frequency

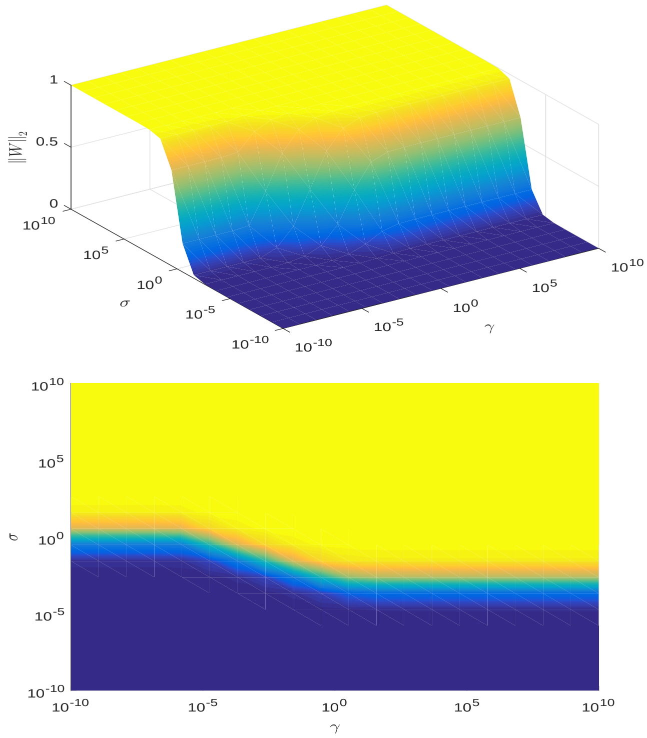


Figure 2.6: $\|\boldsymbol{W}\|_2$ is calculated for a wide range of σ and γ. (Top) As can be seen, the values are bounded between 0 and 1. (Bottom) Upper view of the top figure.

where the original zero was. In other words there is a tradeoff in this example: on one hand the higher the noise the more difficult it is to obtain a good modeling. On the other hand, the lower the noise, the more problematic the zero at 8926Hz becomes. Surprisingly, the noise itself acts as a sort of protection when modeling systems with this type of zeros.

In order to offer another perspective on the way the proposed methodology works, the following example (i.e. Example 2) was used with nonlinear block:

$$x(t) = u(t)\cos\left(u(t)\right) \text{ with } u(t) \in [-15, 15], \tag{2.22}$$

and linear block:

$$y(t) = \frac{B_2(q)}{A_2(q)}x(t) \tag{2.23}$$

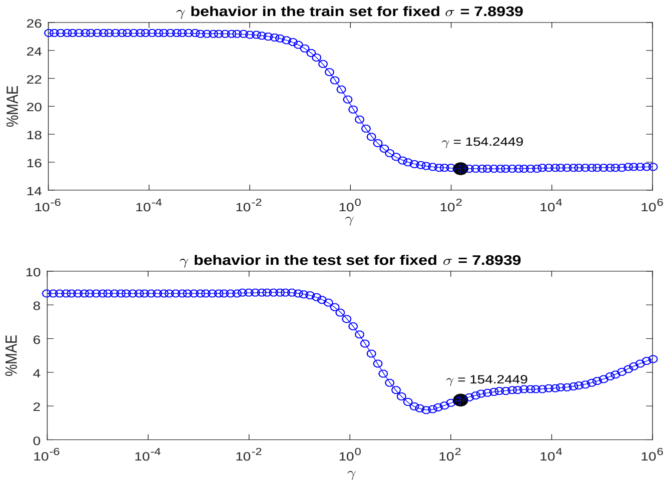


Figure 2.7: σ is fixed while a wide range of γ values is evaluated. The black point corresponds to the chosen value of γ.

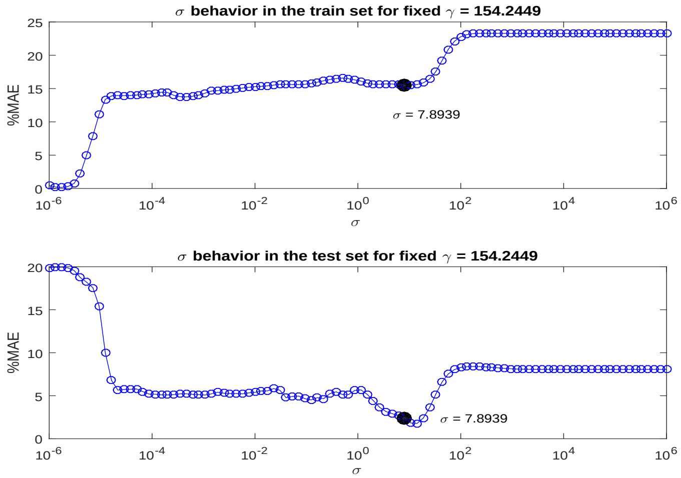


Figure 2.8: γ is fixed while a wide range of σ values is evaluated. The black point corresponds to the chosen value of σ.

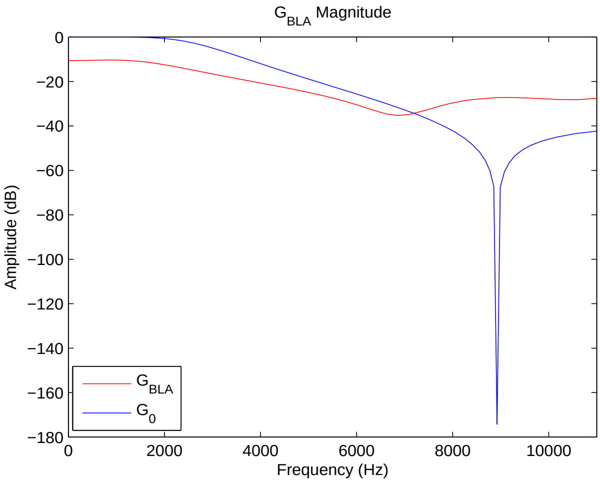


Figure 2.9: Estimated Parametric BLA.

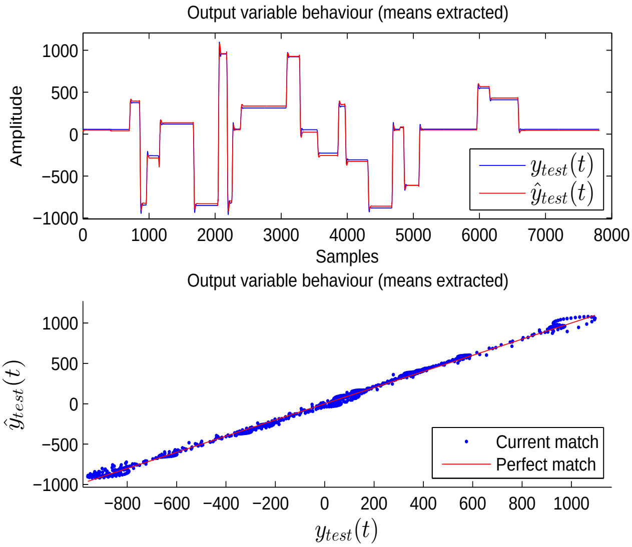


Figure 2.10: (Top) $\hat{y}_{test}(t)$ (rescaled) is superimposed to the actual $y_{test}(t)$. (Bottom) A scatter plot between $\hat{y}_{test}(t)$ and $y_{test}(t)$.

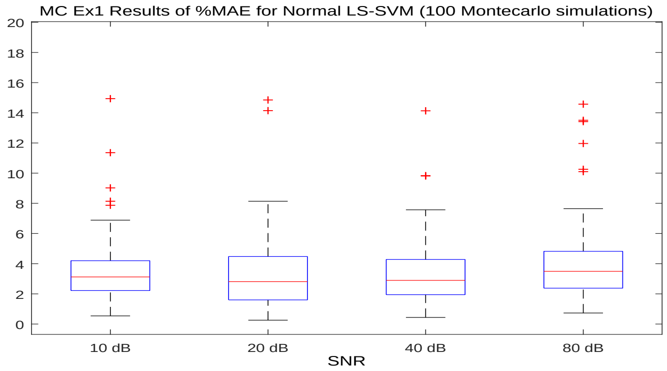


Figure 2.11: Normalized Mean Absolute Error (%MAE) for a 100 Monte Carlo simulation for different levels of noise in Example 1.

where

$$B_2(q) = 0.004728q^3 + 0.01418q^2 + 0.01418q + 0.004728 \tag{2.24}$$

and

$$A_2(q) = q^3 - 2.458q^2 + 2.262q - 0.7654. \tag{2.25}$$

The linear and nonlinear blocks of Example 2 are depicted in Fig. 2.13. Notice that the nonlinearity in this example is particularly difficult to model with polynomial basis functions approaches.

For this example, again 100 Monte Carlo simulations were run and the corresponding results are presented in Fig. 2.14. It can be seen that the results here are more intuitive in the sense that as the SNR increases, the Normalized MAE decreases.

2.3.4 Methods comparison

In order to compare the current method four additional methodologies were considered. The compared methods include:

- Inversion + LS-SVM (i.e. the presented method).
- NARX LS-SVM (Suykens et al., 2002) with 10 lags of input and 10 lags of output.

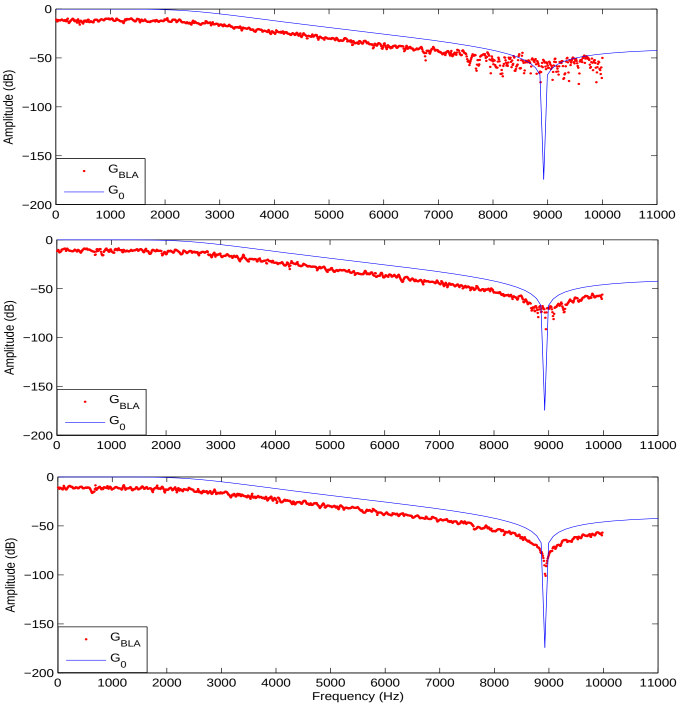


Figure 2.12: Nonparametric BLA for different noise levels: (Top) SNR = 20dB. (Center) SNR = 40dB. (Bottom) SNR = 80dB.

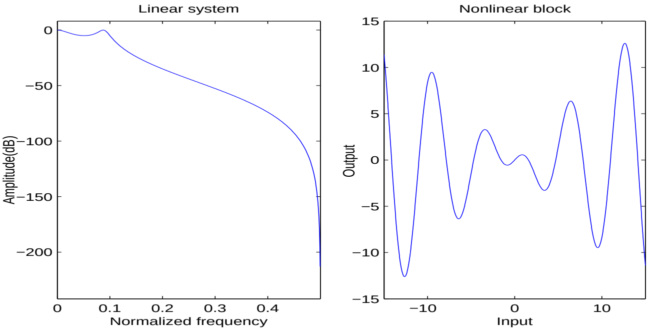


Figure 2.13: (Left) Linear system in normalized frequency and dB. (Right) Nonlinear block.

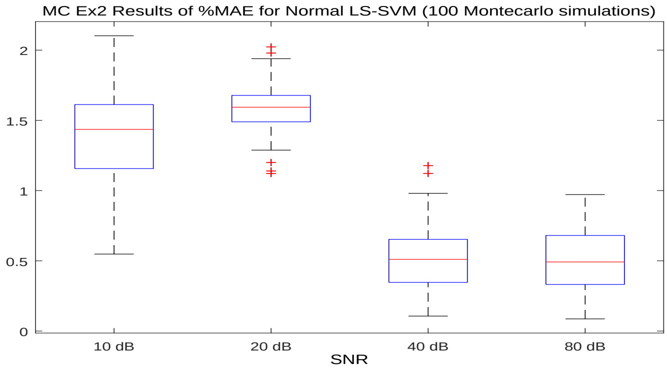


Figure 2.14: Normalized Mean Absolute Error (%MAE) for a 100 Monte Carlo simulation for different levels of noise in Example 2.

- The Hammerstein and Wiener Identification procedure (in this chapter denoted by WHIP) presented in M. Schoukens (2015).
- The iterative method (in this chapter denoted by IM) presented in Bai and Li (2010).
- The State-Dependent Parameter (SDP) method in combination with the RIVBJ routine contained in the CAPTAIN toolbox (P. Young (2000); P. C. Young, McKenna, and Bruun (2001)). This will be referred to as the Captain method.

To carry out the comparison, a third example of a more realistic nature is introduced (from now on referred to as Example 3). This example models a push-pull type B amplifier as depicted in Fig. 2.15. To model the speaker, its electrical and mechanical dynamics were considered. The way the speaker is modeled is represented in Fig. 2.16 where the input is the voltage from the amplifier (i.e. $v(t)$) and the output is the displacement of the cone of the speaker $x_d(t)$. For this example the number of realizations used was 50.

The system is represented as shown in Fig. 2.17 where the nonlinear block is a piecewise function with saturations given by the positive and negative supply rails (i.e. see Fig. 2.15) and a deadzone between -0.55V and 0.55V generated by the transistors. The speaker on the other hand is modeled through a transfer function (in continuous time) as shown in (2.26) (Ravaud, Lemarquand, & Roussel, 2009):

$$\frac{X_d(s)}{V(s)} = \frac{k_m}{(Ls+R)(ms^2+bs+k)-k_vk_ms}, \tag{2.26}$$

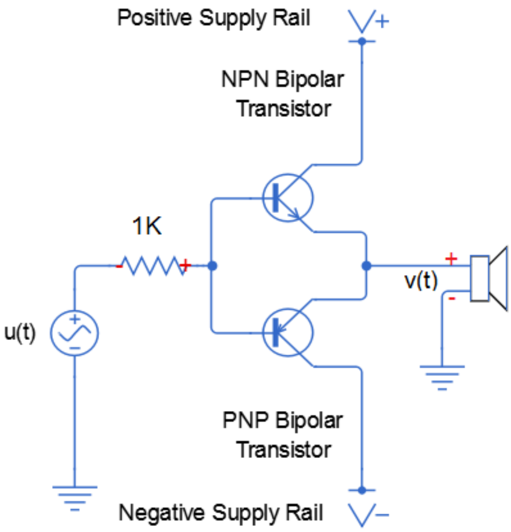


Figure 2.15: Example 3. Push-pull type B amplifier.

where $k_m = 3.14$N/A is the constant from the Lorentz force, $k_v = 3.14$Vs/m is the constant of the electromotive force (i.e. EMF), $k = 20000$N/m is the spring constant, $b = 50$N/m/s is the dampener constant, $m = 0.004$Kg is the mass of the cone and the coil, $R = 5\Omega$ is the resistance and $L = 50\mu H$ the inductance of the speaker. The transfer function was then discretized with the zero order hold method with a sampling time $Ts = 0.0004$s producing finally:

$$x_d(t) = \frac{0.003639q^2 + 0.0009408q + 9.87 \times 10^{-08}}{q^3 - 0.8595q^2 + 0.005371q - 4.302 \times 10^{-20}} v(t). \qquad (2.27)$$

For Examples 1 and 2 the inputs of the test set range between -10 and 10. For Example 3, the test set range of the inputs goes from -20 to 20.

To train the NARX LS-SVM method, uniformly distributed white noise inputs were used. In Examples 1 and 2, $u_{\text{NARX}}(t)$ covers at least $[-15, 15]$ while in Example 3, the amplitude of the input $u_{\text{NARX}}(t)$ covered $[-20, 20]$.

For the training of the Captain method, uniformly distributed white noise inputs were used. In Example 2, $u_{\text{CAP}}(t)$ was scaled so that it covered at least $[-15, 15]$ while in Example 3, the amplitude of the input $u_{\text{CAP}}(t)$ covered $[-20, 20]$. Linear model orders between 2 and 4 were scanned and nonlinear degrees between 3 and 25 for the first and

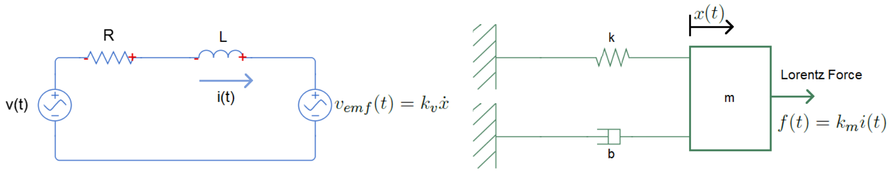


Figure 2.16: Example 3. Speaker modeling. (Left) Electrical dynamics. (Rigth) Mechanical dynamics.

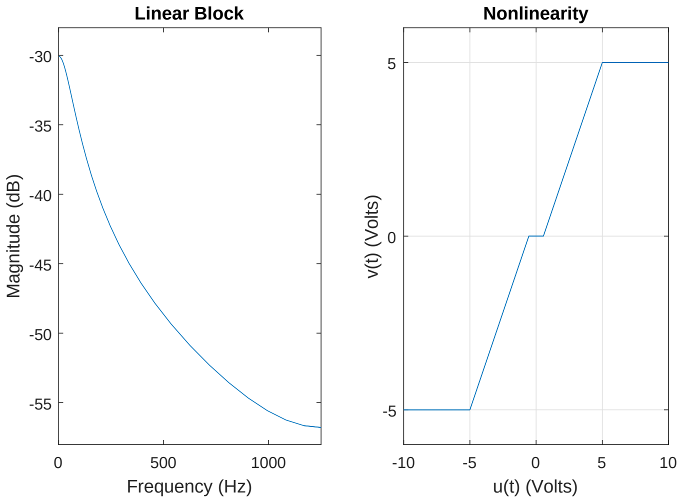


Figure 2.17: Example 3. (Left) Linear system in normalized frequency and dB. (Right) Nonlinear block. For visualization purposes it is only displayed from -10V to 10V while the actual range is from -20 to 20.

second examples and between 3 and 15 for the third one were scanned. The specific functions employed were `sdp` and `rivbj`.

To train the IM method, Gaussian noise inputs were used. In the first example, $u_{\mathrm{IM}}(t)$ had an RMS value of 0.3. In the second example, the amplitude of the Gaussian noise input $u_{\mathrm{IM}}(t)$ was rescaled so that it covers at least $[-15, 15]$. Similarly, in Example 3 $u_{\mathrm{IM}}(t)$ covers at least $[-20, 20]$. This was done to avoid extrapolation issues as the nonlinearity in (2.22) is not in the polynomial model class. These signals were used to estimate iteratively the linear and the nonlinear blocks. Linear model orders between 2 and 4 were scanned. Also, nonlinear degrees between 3 and 5 for the first example, between 3 and 25 for the second one and between 3 and 21 for Example 3 were scanned. The linear model order and the nonlinear degree are chosen simultaneously using cross-validation with the Multilevel Pseudo Random validation signal $u_3(t)$.

For the training of the WHIP case, for Example 1 multisine signals with an RMS value of 0.3 were used as inputs. For Example 2 $u_{\mathrm{WHIP}}(t)$ covered at least $[-10, 10]$. For Example 3, $u_{\mathrm{WHIP}}(t)$ covered $[-20, 20]$. Two steady-state periods of four out of the five phase realizations of the multisine were used for estimation of the linear and the nonlinear blocks. One steady-state period of the fifth realization of the multisine was used for model order selection of the linear block (i.e. the model order was chosen between 2/2, 2/3, 2/4, 3/3, 3/4, and 4/4). A Multilevel Pseudo Random Validation signal $u_3(t)$ was used for selecting the nonlinear degree (i.e. between 3 and 5 for the first example and between 3 and 25 for the second and third ones). After following the steps above, the obtained model was optimized using the four phase realizations of the multisine that were used earlier to estimate the linear and the nonlinear block.

In Table 2.3.4 the results of the comparison in Normalized MAE form are presented. Each of the presented results corresponds to an average over 10 runs.

	Example 1		**Example 2**		**Example 3**	
SNR (dB)	**10**	**20**	**10**	**20**	**10**	**20**
Inversion + LS-SVM	4.3331	4.3735	**0.89992**	**0.76626**	**0.82356**	**0.31645**
NARX LS-SVM	6.2704	11.8499	20.8274	27.0158	3.9029	2.4138
IM	6.6991	6.7313	5.6814	6.1720	1.8026	2.6827
WHIP	**0.0676**	**0.0516**	5.0310	4.9834	1.5271	1.5004
Captain	3.8758	3.4821	12.815	12.1707	26.1013	25.4655

Table 2.2: Results comparison in Normalized MAE.

The IM method seems to be very sensitive to the application of noise. Given that the selected examples contain a low SNR, the performance of the method is severely affected. However, this method obtains better results if the noise is much smaller (e.g. for 80dB SNR a Normalized MAE of 0.00063% is obtained in Example 1).

The IM, Captain and WHIP methods assume that the nonlinearity can be represented in a basis function expansion form with known basis functions. Note that the nonlinearities in (2.22) and Fig. 2.13 and in Fig. 2.17 are hard to model by a polynomial of reasonable degree for the input ranges used (i.e. see Table 2.1). Nevertheless, polynomials were used for these methods.

Given how sharp the zero at 8926Hz is in the linear block of Example 1, it is clear that this is a particularly challenging problem for the proposed methodology due to the inversion step. Nonetheless as can be seen, the proposed method performs well in the three examples. Even more, for the second and third examples it achieves a better performance than the other methods. The results presented suggest that the proposed method is robust against the amount of noise used and also has a great generalization capability when using different model classes, which clearly is a nice advantage.

It is important to highlight that in simpler cases, like the one presented in Example 1, even though the proposed method works well other methods could be better specially when they belong to the model class. For cases where the nonlinearity becomes challenging (i.e. hard nonlinearities), the proposed method becomes an attractive option due to its flexibility and good performance.

2.4 Conclusions

The described method in this chapter presents a combination of techniques, namely the BLA and LS-SVM, for the identification of Hammerstein systems.

It is shown that the inversion of the estimated linear block can be used to make a preliminary estimation of the intermediate variable even in the presence of measurement noise. With this preliminary estimation and the known input the nonlinear block can be modeled. This is, as long as there is a mechanism to counter the influence of the back-propagated noise. In this chapter, the regularization provided by the LS-SVM methodology provides such a tool. Once this modeling is done, the intermediate variable can be re-estimated straightforwardly.

In Hammerstein systems the estimated intermediate variable, in combination with the known input, is enough to model the nonlinear block. Similarly, the intermediate variable in combination with the output variable can be used to model the linear block. Given this, the proposed method allows us to obtain a model for each of the composing blocks of the Hammerstein system (up to a certain scaling factor) that can reproduce the input-output dynamics in an accurate way. This allows a deeper insight into the inner workings of the studied Hammerstein systems.

The method offered in this chapter offers high flexibility with regard to the model class of the nonlinearity it can handle. Furthermore, when dealing with hard nonlinearities the presented method tends to perform better than other state of the art methods thanks to the generalization and regularization capabilities of LS-SVM.

Further extensions of the methodology could be achieved through its application to the MIMO case of Hammerstein systems and to Wiener-Hammerstein systems.

Chapter 3

Incorporating Best Linear Approximation within LS-SVM-Based Hammerstein System Identification

3.1 Introduction

The objective of this chapter is to incorporate the techniques of the Best Linear Approximation (BLA) (Pintelon & Schoukens, 2012) within Least Squares Support Vector Machines (LS-SVM) (Suykens et al., 2002). In the proposed method it is possible to clearly separate the steps for identification of the linear and nonlinear parts.

Under the proposed methodology, it will be shown that the solution of the model follows from solving a linear system of equations. By itself, this already constitutes an advantage over other methods like overparametrization in the sense that the proposed method is much more simple and easy to implement.

Incorporating information of the system's structure into a LS-SVM model can be difficult. To do that, in this chapter we use the BLA approach to model the linear block

This chapter is based on:
Castro-Garcia, R., Tiels, K., Schoukens, J., Suykens, J. A. K. (2015). *Incorporating Best Linear Approximation within LS-SVM-Based Hammerstein System Identification*. In proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015), Osaka, Japan. (pp. 7392 - 7397).

and use the results to help LS-SVM modeling the nonlinear part. To achieve this, the primal formulation of LS-SVM is modified to include the information of the structure of the system and the approximation to the linear block obtained through the BLA.

The proposed methodology can be separated in two stages:

- The system's BLA is calculated and used as an approximation to the linear block.
- A modified LS-SVM model is trained including the information given by the BLA of the system.

Note then that the full Hammerstein model consists of a nonlinear block given by the resulting LS-SVM model and a linear part coming from the BLA.

The proposed method is applied to two simulation examples and the results are presented. There, the output of the Hammerstein system is measured in the presence of white Gaussian additive noise (i.e. $v(t)$ in Fig. 1.1).

It will be shown that in the presence of noise, the method can very effectively calculate an approximation to the nonlinear model (up to a scaling factor) and to the system as a whole. It is important to highlight that this scaling factor is not identifiable (Boyd & Chua, 1983).

This chapter is organized as follows: In Section 3.2 the problem statement is offered. The proposed method is presented in Section 3.3 where it is explained how the BLA and LS-SVM were used together. Section 3.4 illustrates the results found when applying the described methodology on two simulation examples. Finally, in Section 3.5, the conclusions and ideas for future work are presented.

3.2 Problem Statement

To represent a linear dynamic block, an ARX model can be used (Ljung, 1999):

$$\hat{y}(t) = \sum_{j=0}^{m} b_j u(t-j) - \sum_{i=1}^{n} a_i y(t-i). \tag{3.1}$$

Here, $\hat{y}(t)$ is the currently estimated value of the output, while $y(t-i)$ are past outputs and $u(t-j)$ represents the past and present inputs. Note that b_j and a_i represent the coefficients of the numerator and denominator of the linear block respectively.

In the Hammerstein case, the input $u(t)$ goes trough a nonlinear block first. This nonlinear block is represented as $f(u(t))$ in Fig. 1.1, therefore, the model is expressed as:

$$y(t) = \sum_{j=0}^{m} b_j f(u(t-j)) - \sum_{i=1}^{n} a_i y(t-i) + e(t). \tag{3.2}$$

To obtain an approximation to the coefficients a_i and b_j, the BLA approach will be used and to obtain a representation of $f(u(t))$, LS-SVM will be employed.

3.3 Proposed Method

Given that an approximation to the a_i and b_j coefficients of the transfer function can be estimated from the BLA (i.e. see Appendix A), the aim is to incorporate this approximation in the formulation of LS-SVM to exploit the knowledge of the structure of the system. This gives the following model to be identified:

$$y(t) = \sum_{j=0}^{m} b_j (\boldsymbol{w}^\top \varphi(u(t-j)) + d_0) - \sum_{i=1}^{n} a_i y(t-i) + e(t). \tag{3.3}$$

For this model, one formulates the following constrained optimization problem:

$$\min_{\boldsymbol{w}, d_0, \boldsymbol{e}} J = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\sum_{t=r}^{N} e(t)^2 \tag{3.4}$$

s.t. eq. (3.3) holds for all $t = r, ..., N$. Here $r = \max(n, m) + 1$.

Given these elements, one has the following Lagrangian:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, d_0, \boldsymbol{e}, \boldsymbol{\alpha}) =& J - \sum_{t=r}^{N} \alpha_t \left(\sum_{j=0}^{m} b_j (\boldsymbol{w}^\top \varphi(u(t-j)) + d_0) \right. \\ & \left. - \sum_{i=1}^{n} a_i y(t-i) + e(t) - y(t) \right).\end{aligned} \tag{3.5}$$

The optimality conditions become:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 & \rightarrow \quad \boldsymbol{w} = \sum_{t=r}^{N} \alpha_t \sum_{j=0}^{m} b_j \varphi(u(t-j)) \\ \frac{\partial \mathcal{L}}{\partial d_0} = 0 & \rightarrow \quad \sum_{t=r}^{N} \alpha_t \sum_{j=0}^{m} b_j = 0 \\ \frac{\partial \mathcal{L}}{\partial e_t} = 0 & \rightarrow \quad \alpha_t = \gamma e_t \text{ for } t = r, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_t} = 0 & \rightarrow \quad y(t) = \sum_{j=0}^{m} b_j (\boldsymbol{w}^\top \varphi(u(t-j)) + d_0) \\ & \qquad\qquad - \sum_{i=1}^{n} a_i y(t-i) + e(t) \text{ for } t = r, ..., N. \end{cases} \tag{3.6}$$

By replacing the first and third conditions (i.e. $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0$ and $\frac{\partial \mathcal{L}}{\partial e_t} = 0$) into the last one (i.e. $\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0$) one gets for $t = r, ..., N$:

$$\begin{aligned} y(t) &= \sum_{j=0}^{m} b_j \left(\left(\sum_{q=r}^{N} \alpha_q \sum_{p=0}^{m} b_p \varphi(u(q-p)) \right)^\top \right. \\ &\quad \left. \varphi(u(t-j)) + d_0 \right) - \sum_{i=1}^{n} a_i y(t-i) + \frac{\alpha_t}{\gamma} \\ &= \sum_{j=0}^{m} \sum_{q=r}^{N} \sum_{p=0}^{m} b_j b_p \alpha_q \varphi(u(q-p))^\top \varphi(u(t-j)) \\ &\quad + \sum_{j=0}^{m} b_j d_0 - \sum_{i=1}^{n} a_i y(t-i) + \frac{\alpha_t}{\gamma}. \end{aligned} \tag{3.7}$$

Let us define:

$$\eta = N - r + 1 \tag{3.8}$$

$$\tilde{b} = \sum_{j=0}^{m} b_j \tag{3.9}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_r & \cdots & \alpha_N \end{bmatrix}^\top \quad \in \mathbb{R}^{\eta} \tag{3.10}$$

$$\boldsymbol{a} = -\begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}^\top \quad \in \mathbb{R}^{n} \tag{3.11}$$

$$\boldsymbol{y}_f = \begin{bmatrix} y_r & \cdots & y_N \end{bmatrix}^\top \quad \in \mathbb{R}^{\eta} \tag{3.12}$$

$$\boldsymbol{\Omega}_{k,l} = \varphi(u_k)^\top \varphi(u_l) \quad \text{for } k, l = 1, ..., N \tag{3.13}$$

$$\boldsymbol{M}_{q,t} = \sum_{j=0}^{m} \sum_{p=0}^{m} b_j b_p (\boldsymbol{\Omega}_{(q-p,t-j)}) \quad \text{for } t, q = r, ..., N \tag{3.14}$$

$$\boldsymbol{Y}_p = \begin{bmatrix} y_{r-1} & y_r & \cdots & y_{N-1} \\ y_{r-2} & y_{r-1} & \cdots & y_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{r-n} & y_{r-n+1} & \cdots & y_{N-n} \end{bmatrix} \quad \in \mathbb{R}^{n \times \eta} \tag{3.15}$$

From $\frac{\partial \mathcal{L}}{\partial d_0} = 0$ one gets

$$\sum_{t=r}^{N} \alpha_t \tilde{b} = \tilde{b}\mathbf{1}^\top \boldsymbol{\alpha} = 0 \tag{3.16}$$

and from $\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0$:

$$\begin{aligned} \boldsymbol{y}_f &= \sum_{j=0}^{m}\sum_{p=0}^{m} b_j b_p \boldsymbol{\Omega}_{(q-p,t-j)}\boldsymbol{\alpha} + \boldsymbol{Y}_p^\top \boldsymbol{a} + \tilde{b}\mathbf{1}^\top d_0 + \gamma^{-1}\boldsymbol{I}\boldsymbol{\alpha} \\ &= \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{Y}_p^\top \boldsymbol{a} + \tilde{b}\mathbf{1}^\top d_0 + \gamma^{-1}\boldsymbol{I}\boldsymbol{\alpha}, \end{aligned} \tag{3.17}$$

with $t, q = r, ..., N$.

The obtained linear system can now be written as:

$$\begin{bmatrix} 0 & \tilde{b}\mathbf{1}_\eta^\top \\ \tilde{b}\mathbf{1}_\eta & (\boldsymbol{M} + \frac{\boldsymbol{I}}{\gamma}) \end{bmatrix} \begin{bmatrix} d_0 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y}_f - \boldsymbol{Y}_p{}^\top \boldsymbol{a} \end{bmatrix}. \tag{3.18}$$

Under this representation, the model is linear in the unknowns and therefore it can be solved directly.

Note that once $\boldsymbol{\alpha}$ and d_0 are known, it is possible to directly apply the model to new data points.

3.4 Results

The proposed methodology was applied to two systems in the discrete time framework. In order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D. The first system was generated through a nonlinear block:

$$x(t) = u(t)^3 \tag{3.19}$$

and a linear block:

$$y(t) = \frac{0.004728q^3 + 0.01418q^2 + 0.01418q + 0.004728}{q^3 - 2.458q^2 + 2.262q - 0.7654} x(t). \tag{3.20}$$

The second system was generated through a nonlinear block:

$$x(t) = -0.5u(t)^3 + 5u(t)^2 + u(t), \tag{3.21}$$

Table 3.1: Selected Parameters

	σ	γ
Ex. 1	102.4343	42.4872
Ex. 2	398.7231	3.9057×10^4

and a linear block:

$$y(t) = \frac{0.1129q^4 - 0.2128q^3 + 0.283q^2 - 0.2128q + 0.1129}{q^4 - 2.485q^3 + 2.528q^2 - 1.184q + 0.2245} x(t). \tag{3.22}$$

For the first system a ramp signal from $\{-15, 15\}$ and slope of 45 degrees was used for training. In example 2 this same ramp was used but its values were shuffled. For the test set of both systems a Multi Level Pseudo Random Signals (MLPRS) with an amplitude $\in \{-10, 10\}$ and a switching probability of 2% was used. For the tuning of the LS-SVM parameters (i.e. σ and γ) a coupled simulated annealing (CSA) Xavier-de Souza et al. (2009) followed by simplex Nelder and Mead (1965) was used under a 10-fold crossvalidation scheme. For the first example, the training data set consisted of 2000 points, while the test set consisted of 2500 data points. For the second example 1000 points were used for both training and testing. The corresponding selected values are shown in Table 3.1 for 40dB of signal to noise ratio (SNR).

The results for the first example can be seen in Figs. 3.1 and 3.2 while the results for the second example are shown in Figs. 3.3 and 3.4. In Figs. 3.1 and 3.3 the mean values of $y(t)$ and $\hat{y}(t)$ were extracted and their Normalized Mean Absolute Error (%MAE) was calculated. Figs. 3.2 and 3.4 show the comparison between the estimated nonlinearities and the real ones. It is evident that even though they have very different magnitudes, their shape is quite similar. Note that this difference in scaling points to a factor appearing between the two blocks of the system.

Both systems were affected with white Gaussian noise. Here however, unlike the BLA on the estimation, only a single realization was used.

The method was able to retrieve good approximations despite the noise. Figs. 3.5 and 3.6 show the evolution of the distributions of deviations from the actual output as the SNR changes.

As can be seen, the distribution of deviations from the actual output broadens as the SNR decreases. However, even with a large presence of noise, smaller deviations are more frequent which is in line with the type and magnitude of the noise introduced in the measurements.

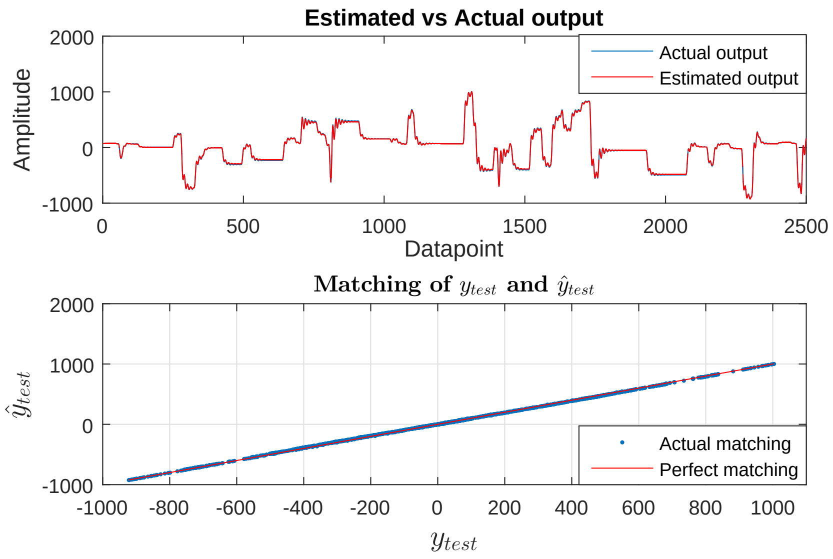


Figure 3.1: Example 1: Overlapping of the actual output variable y and the estimation $\hat{y}$. Means extracted, $\%\text{MAE} = 0.33792\%$

Table 3.2: %MAE comparison for Example 1

SNR	10	20	30	40
BLA+LS-SVM	0.9167	0.3496	0.3345	0.3379
NARX LS-SVM	3.2058	1.7367	0.8899	0.4410

Given that the proposed method takes the underlying structure of the system into account, it should better model the system than purely black box methods. Tables 3.2 and 3.3 show the results of the comparison between the proposed method and a NARX LS-SVM model in the test set. These results were obtained when applying different SNR values to examples 1 and 2 respectively. It can be seen that the proposed method clearly outperforms the purely black box approach of NARX LS-SVM. For the NARX LS-SVM 2000 training points were used for the first example and 1000 for the second. Also, 10 lags of input and 10 lags of output were employed.

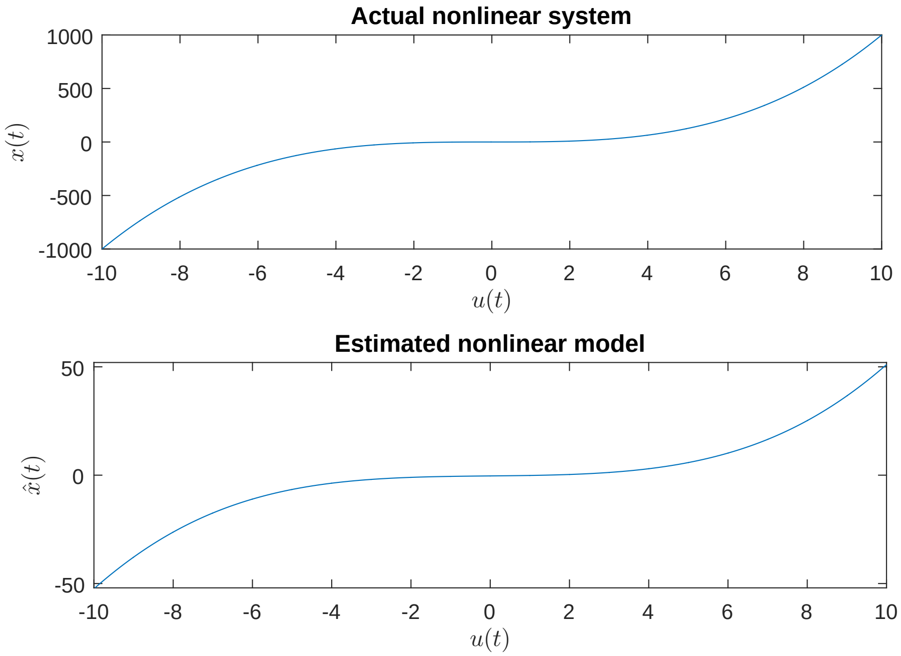


Figure 3.2: Example 1: Comparison between the actual nonlinear system and the estimated model

Table 3.3: %MAE comparison for Example 2

SNR	10	20	30	40
BLA+LS-SVM	8.1497	1.9835	0.5015	0.0926
NARX LS-SVM	6.5407	5.0459	1.8062	1.8882

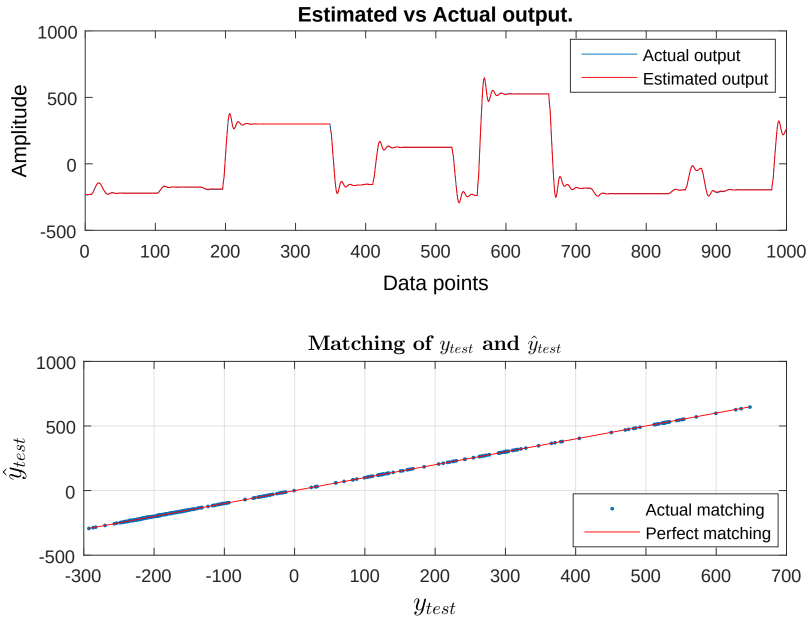


Figure 3.3: Example 2: Overlapping of the actual output variable y and the estimation $\hat{y}$. Means extracted, $\%\mathrm{MAE} = 0.092553$

3.5 Conclusions

The method presented in this chapter combines two powerful techniques, namely LS-SVM and BLA, which when used in combination turn out to be quite effective for the identification of Hammerstein systems. In particular, the estimation of the linear block from BLA was used in the formulation of the dual representation for estimating the LS-SVM model.

The results presented indicate that the method is very effective in the presence of zero mean, white Gaussian noise. For this method, the kernel parameter σ and the regularization parameter γ have to be tuned. However, once the model is learned, it can be easily applied. It is important to highlight that the estimated nonlinear model is very close to the original one up to a scaling factor, which allows great insight into the behavior of the system studied.

The solution of the model follows from solving a linear system of equations, which

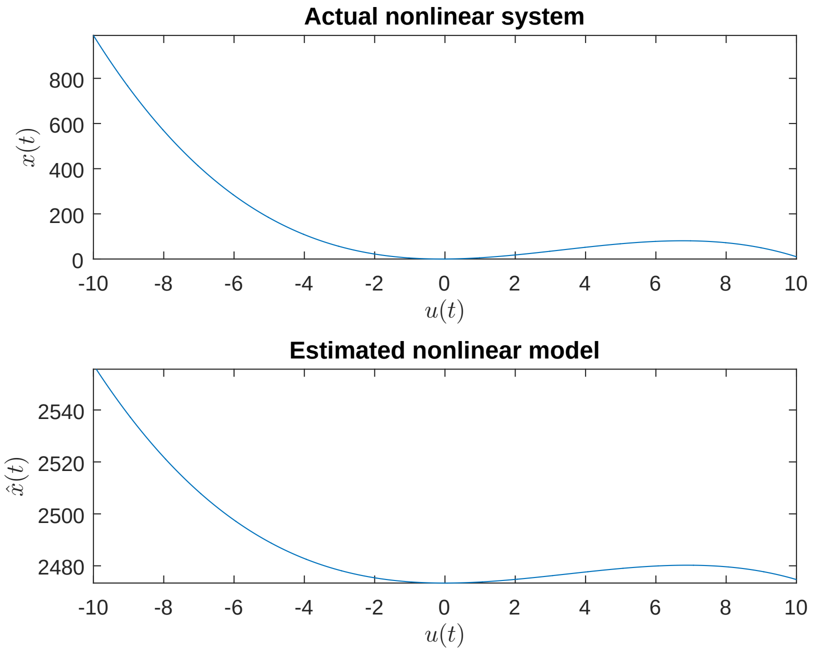


Figure 3.4: Example 2: Comparison between the actual nonlinear system and the estimated model

constitutes an advantage over other methods like the overparametrization presented in Goethals et al. (2005). This is, in the sense of how easy it is to solve these equations and afterwards to apply the found model.

Future work for the presented method could includes the extension of the method to other block oriented structures like Wiener-Hammerstein systems where, after the identification of the estimated input and output linear blocks, the method could be applied. To separate these blocks, for example the phase coupled multisine approach (J. Schoukens, Tiels, & Schoukens, 2014) could be used.

Generalizing this method to express the cost function in the frequency domain would allow one to focus the fit of the model in a specifically needed part of the frequency band.

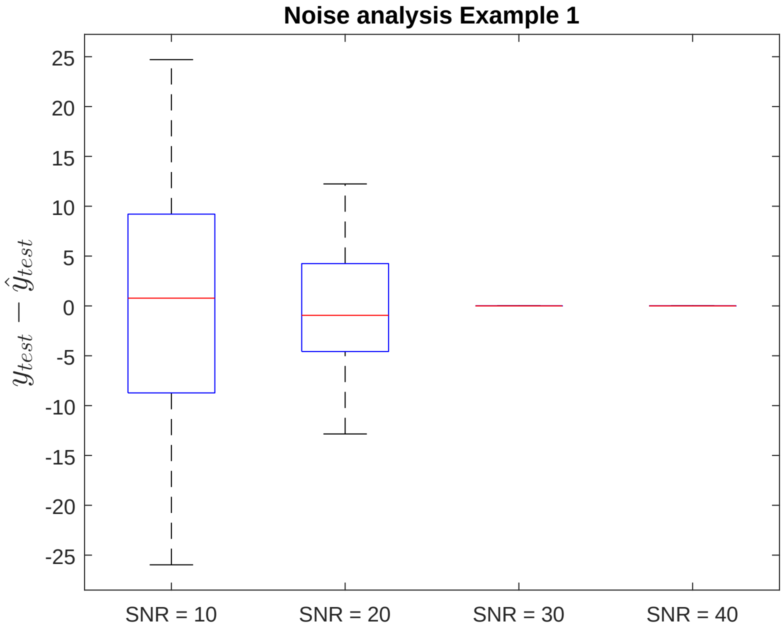


Figure 3.5: Example 1: evolution of the distributions of deviations from the actual output as the SNR changes.

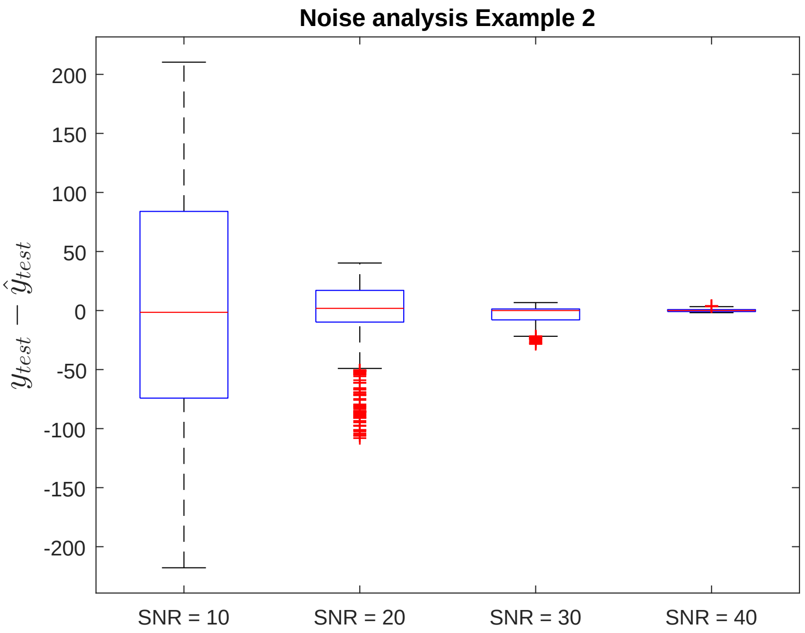


Figure 3.6: Example 2: evolution of the distributions of deviations from the actual output as the SNR changes.

Chapter 4

Wiener System Identification using Best Linear Approximation within the LS-SVM framework

4.1 Introduction

The objective in this chapter is to incorporate the techniques of the Best Linear Approximation (BLA) (Pintelon & Schoukens, 2012) within Least Squares Support Vector Machines (LS-SVM) (Suykens et al., 2002) for the identification of Wiener Systems. It will be assumed that the intermediate variable between the two blocks is unknown, this is: only the input and output can be sampled.

The incorporation of additional information regarding the structure of the system into an LS-SVM model can be difficult. In this chapter the BLA approach is used to model the linear block and these results are used to help LS-SVM modeling the nonlinear part. For the proposed method it will be shown that the solution of the model follows from solving a linear system of equations. By itself, this already constitutes

This chapter is based on:
Castro-Garcia, R., Suykens, J. A. (2016). *Wiener System Identification using Best Linear Approximation within the LS-SVM framework.* In proceedings of the 3rd Latin American Conference on Computational Intelligence. doi:10.1109/LA-CCI.2016.7885698.

an advantage over other methods like overparametrization given the simplicity and easiness of implementation while offering a very good performance.

The proposed methodology can be separated in four stages:

- The system's BLA is calculated.
- A parametric version of the BLA is estimated and used as an approximation to the linear block.
- An approximation to the intermediate variable $\hat{x}(t)$ is obtained using the parametric BLA and the known input $u(t)$.
- An LS-SVM model is trained using $\hat{x}(t)$ and the known $y(t)$.

Note then that the full Wiener model consists of a linear part coming from the BLA and a nonlinear block given by the resulting LS-SVM model.

In this chapter, the method is applied to two simulation examples and the results are presented. In the examples, the output of the Wiener system is measured in the presence of white Gaussian additive noise (i.e. $v(t)$ in Fig. 1.2). It is shown that also in the presence of noise, the method can very effectively calculate an approximation to the system as a whole.

This chapter is organized as follows: In Section 4.2 the problem statement is offered. The proposed method is presented in Section 4.3 where it is explained how the BLA and LS-SVM were used together. Section 4.4 illustrates the results found when applying the described methodology on two simulation examples. Finally, in Section 4.5, the conclusions and ideas for future work are presented.

4.2 Problem Statement

In the Wiener case, the input $u(t)$ goes through a linear block first. To represent a linear dynamic block, an ARX model can be used (Ljung, 1999):

$$x(t) = \sum_{j=0}^{m} b_j u(t-j) - \sum_{i=1}^{n} a_i x(t-i). \tag{4.1}$$

Here $x(t)$ is the intermediate variable at time t, while $x(t-i)$ are past outputs of such model and $u(t-j)$ the past and present inputs.

After this, the intermediate variable goes through a nonlinear block. This block is represented as $f(x(t))$ in Fig. 1.2, therefore, the output $\hat{y}(t)$ can be represented as:

$$\hat{y}(t) = f(x(t)) = f\left(\sum_{j=0}^{m} b_j u(t-j) - \sum_{i=1}^{n} a_i x(t-i)\right). \tag{4.2}$$

To obtain a representation of the coefficients a_i and b_j, the BLA approach will be used and to obtain an approximation to $f(x(t))$, LS-SVM will be employed.

4.3 Proposed Method

The goal of this chapter is to incorporate the coefficients estimated through the BLA (i.e. $\hat{a}_i$ and $\hat{b}_j$) into an LS-SVM model to exploit the knowledge of the structure of the system. With these coefficients, obtaining an approximation to the intermediate variable $\hat{x}(t)$ is straightforward:

$$\hat{x}(t) = \sum_{j=0}^{m} \hat{b}_j u(t-j) - \sum_{i=1}^{n} \hat{a}_i \hat{x}(t-i). \tag{4.3}$$

Replacing (4.3) into (4.2) we get $\hat{y}(t) = f(\hat{x}(t))$ and using (1.28) to model the nonlinear block, we can estimate an approximation to the output signal $\hat{y}(t)$:

$$\hat{y}(t) = \boldsymbol{w}^\top \boldsymbol{\varphi}(\hat{x}(t)) + d. \tag{4.4}$$

For this model, one formulates the following constrained optimization problem:

$$\min_{\boldsymbol{w}, d_0, \boldsymbol{e}} J = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\sum_{t=1}^{N} e^2(t) \tag{4.5}$$

s.t. eq. (4.4) holds for all $t = 1, ..., N$.

Given these elements, one has the following Lagrangian:

$$\mathcal{L}(\boldsymbol{w}, d, \boldsymbol{e}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\sum_{t=1}^{N} e^2(t) - \sum_{t=r}^{N} \boldsymbol{\alpha}_t(\boldsymbol{w}^\top \boldsymbol{\varphi}(\hat{x}(t)) + d + e(t) - y(t)). \tag{4.6}$$

From here, it is evident that the formulation becomes exactly as described in Section 1.7.1. This is very convenient as the problem becomes then a standard LS-SVM one after we obtain the coefficients $\hat{a}_i$ and $\hat{b}_i$.

Note that the order of the transfer function representing the linear block is unknown, this means that different order values have to be tried until a fitting combination is found.

It is important to note that there is a scaling factor that differentiates the actual $G_0(q)$ and the actual $G_{BLA}(q)$. This scaling difference implies that the nonlinear model will have to compensate for the difference so it has no effect on the input-output behavior of the estimated Wiener model (i.e. any pair of $\{G(q)/\eta,\ \eta f(x(t))\}$ with $\eta \neq 0$ would yield identical input and output measurements). However, this factor between the blocks is unidentifiable (Boyd & Chua, 1983).

The accuracy of the method will depend then on how well the parameters of the linear block are estimated as will be shown in Section 4.4.2.

4.4 Simulation Results

4.4.1 Examples

The proposed methodology was applied to two systems in the discrete time framework. In order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D. The first system was generated through a nonlinear block:

$$y(t) = x(t)^3 \tag{4.7}$$

and a linear block:

$$x(t) = \frac{B_1(q)}{A_1(q)}u(t) \tag{4.8}$$

where

$$\begin{aligned} B_1(q) = &\quad 0.0089q^3 - 0.0045q^2 - 0.0045q + 0.0089 \\ A_1(q) = &\quad q^3 - 2.5641q^2 + 2.2185q - 0.6456. \end{aligned} \tag{4.9}$$

The second system was generated through a nonlinear block:

$$y(t) = \text{sinc}(x(t))x(t)^2 \tag{4.10}$$

and a linear block:

$$x(t) = \frac{B_2(q)}{A_2(q)}u(t) \tag{4.11}$$

where

$$\begin{aligned} B_2(q) = &\quad 0.0047q^3 + 0.0142q^2 + 0.0142q^1 + 0.0047 \\ A_2(q) = &\quad q^3 - 2.458q^2 + 2.262q - 0.7654 \end{aligned} \tag{4.12}$$

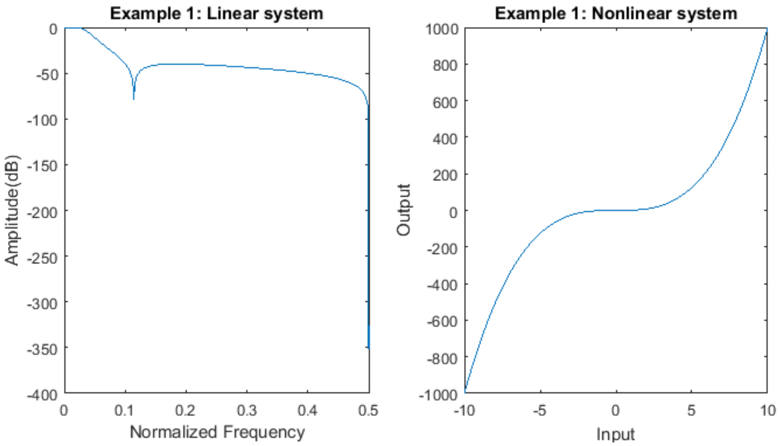


Figure 4.1: Example 1. (Left) Linear system. (Right) Nonlinear system.

Figures 4.1 and 4.2 illustrate examples 1 and 2 respectively. Both systems were trained using a ramp signal from -15 to 15 (45 degrees slope). Also, in both cases the test sets were Multi Level Pseudo Random Signals (MLPRS) with an amplitude $\in \{-10, 10\}$. A Coupled Simulated Annealing algorithm was used to tune the parameters (i.e. σ and γ) using a 10-fold Cross-Validation scheme (e.g. LS-SVMlab v1.8). The training set for the nonlinear block and the test data set consisted each of 1000 points.

For the estimation of the orders of the transfer function, values from $n \in \{1, 5\}$ and $m \in \{1, 5\}$ with $n \geq m$ were tried (i.e. see (4.1)). At each iteration, the combination of n and m giving the best accuracy was selected.

Results for Example 1 can be seen in Figs. 4.3, 4.4 and 4.5 and results for Example 2 are shown in Figs. 4.6, 4.7 and 4.8. The systems corresponding to both examples were affected with white Gaussian noise (i.e. A Signal to Noise Rartio of 40dB was used in Figures 4.3 to 4.8).

Figures 4.3 and 4.6 show the estimated model of the linear blocks from the BLA for both examples. Note that the perturbation in the non-paranetric G_{BLA} after 33% of the frequency is due to the lack of excitation from the used signals. Figures 4.4 and 4.7 show the behavior of the estimated model of the nonlinear block for the training set of both examples. Finally, Figures 4.5 and 4.8 show the behavior of the estimated model of the whole system in the test set for each example.

Note that even though the models of the linear an nonlinear parts have different

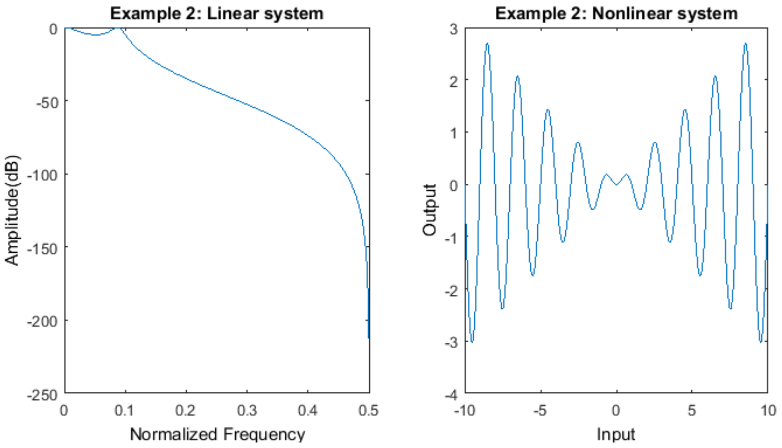


Figure 4.2: Example 2. (Left) Linear system. (Right) Nonlinear system.

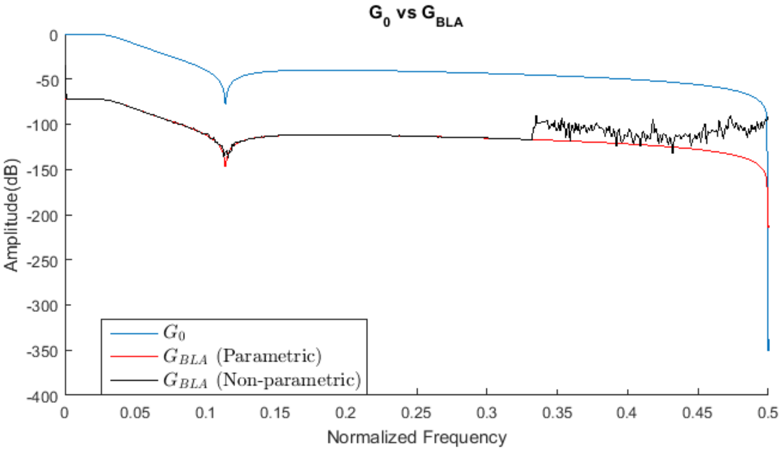


Figure 4.3: Example 1. Linear block estimation.

magnitudes than their corresponding actual blocks, their shape is very similar. The difference in scaling in the linear and nonlinear blocks points to a factor appearing between the two blocks of the system. This factor is unidentifiable and can be distributed between the two blocks as mentioned in Section 4.3.

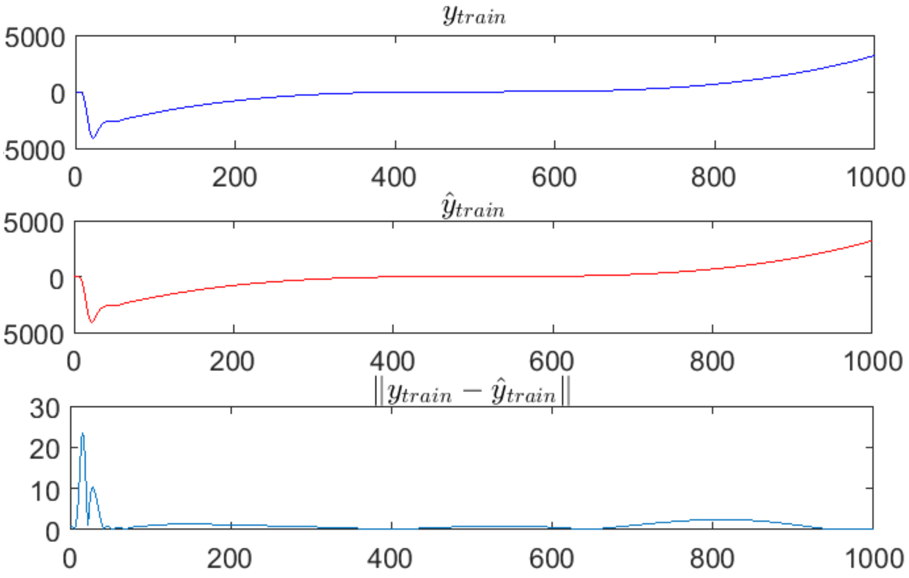


Figure 4.4: Example 1. Non-Linear block behavior in the training set. Horizontal axes are the samples. Vertical axes are amplitude. (Top) Actual training output. (Middle) Estimated train output. (Bottom) Difference between actual and estimated outputs.

For the estimation of the BLA multiple realizations were used (i.e. 5000 realizations of 1000 points each for each example). This diminishes the effect of the noise considerably in the linear block modeling. Further study of the impact of the number of points used during the BLA estimation and the effect of different levels of noise will be considered next.

4.4.2 Impact of number of realizations for the BLA

In order to determine the effect of the number of realizations and the number of points per realization for the estimation of the BLA and subsequently for the accuracy of the model, a series of Monte Carlo simulations were run. For every different number of realizations and points per realizations used, 20 Monte Carlo simulations were carried out and the average of their Normalized MAEs is presented in Table 4.1. Three different levels of Signal to Noise Ratio were used to offer a view of how the relevance of this options vary with the level of noise present.

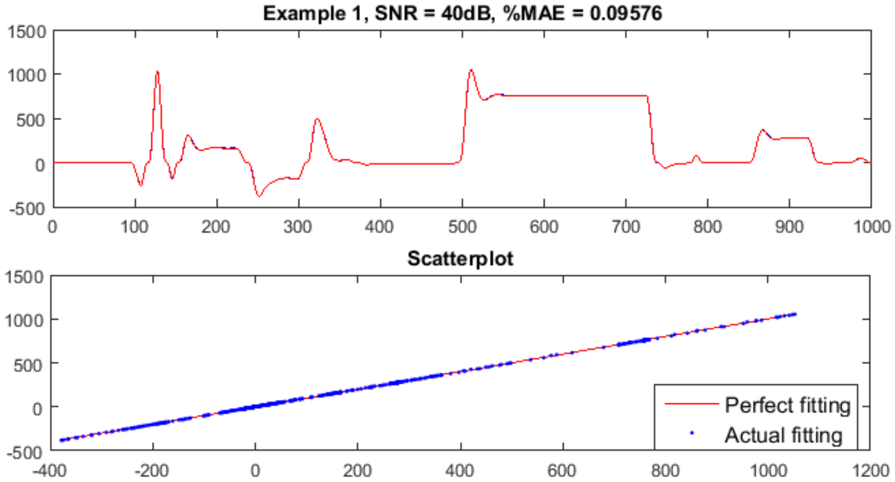


Figure 4.5: Example 1. Model behavior in the test set. (Top) Overlapping of actual $y_{test}(t)$ and $\hat{y}_{test}(t)$. (Bottom) Scatterplot comparing the ideal and actual outputs.

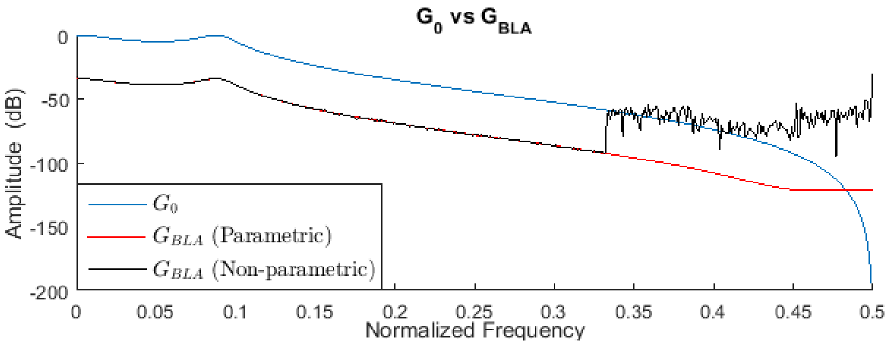


Figure 4.6: Example 2. Linear block estimation.

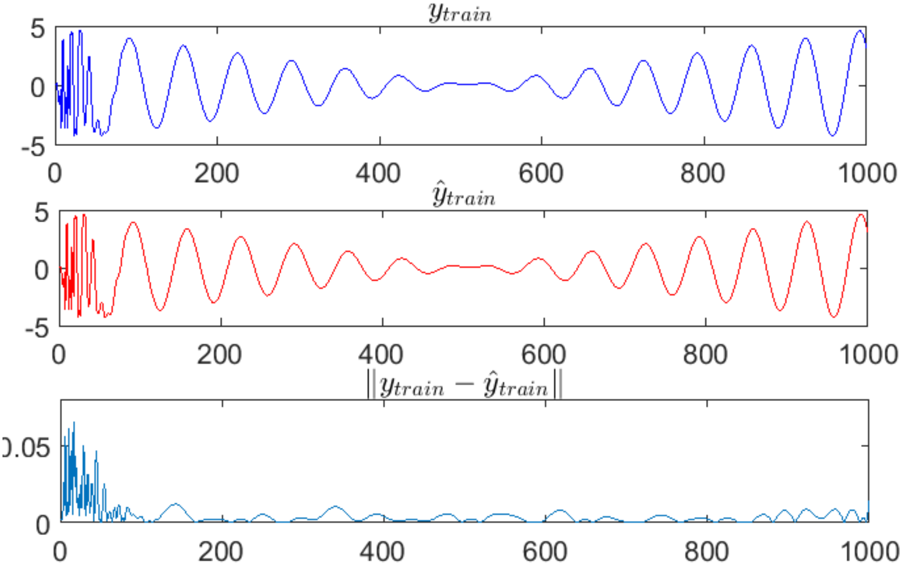


Figure 4.7: Example 2. Non-Linear block behavior in the training set. Horizontal axes are the samples. Vertical axes are amplitude. (Top) Actual training output. (Middle) Estimated train output. (Bottom) Difference between actual and estimated outputs.

		Example 1				Example 2			
		Realizations				Realizations			
SNR	Points	10	100	1000	5000	10	100	1000	5000
Inf	100	24.1393	6.8191	0.24781	0.22055	74.3324	13.1627	5.0186	0.71159
	500	102.4168	0.34096	0.067456	0.046364	15.0309	4.3071	0.35229	0.15282
	1000	63.2599	0.41402	0.041593	0.028606	15.5399	0.91661	0.20108	0.080908
	2000	376.8908	0.4435	0.036867	0.01633	8.1452	0.45106	0.11077	0.037773
40dB	100	134.3097	14.6472	0.2178	0.22376	20.3352	12.2842	3.0184	0.57317
	500	18.8368	0.44293	0.081862	0.062664	15.0489	3.6523	0.31716	0.19719
	1000	48.1492	0.46629	0.07963	0.059179	15.3069	0.95028	0.18943	0.089143
	2000	240.6889	0.50775	0.077693	0.049547	5.7503	0.50761	0.12734	0.089663
10dB	100	1627.2739	1785.4266	1.8413	1.3841	124.9963	11.6749	3.7194	2.599
	500	13.4591	2.4265	1.3931	1.2077	13.3474	4.464	2.1215	1.7422
	1000	12.4578	2.4247	1.241	1.0101	12.8389	2.821	1.7309	1.7859
	2000	11.8653	2.3946	1.1708	1.0846	11.9016	2.1782	1.7085	1.6505

Table 4.1: Effect of the number of points used in the BLA estimation over the $\%MAE$ of the resulting model. Each value reflects the mean of the $\%MAE$ over 20 Monte Carlo Simulations.

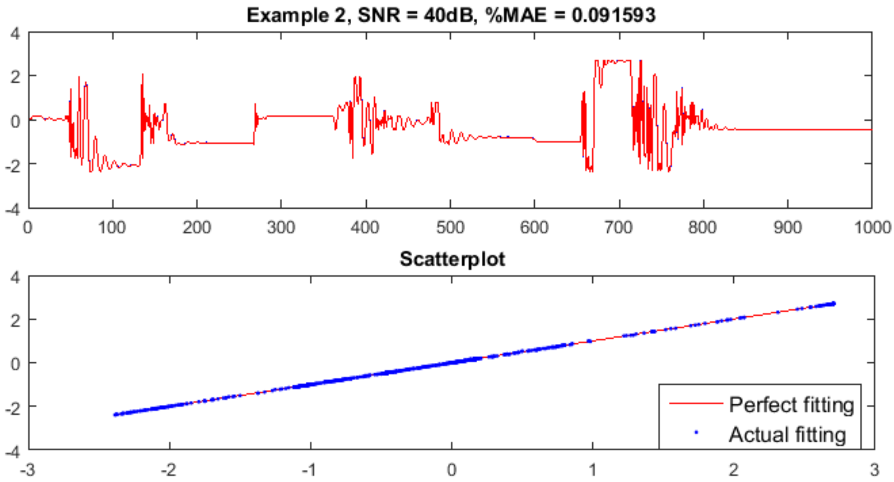


Figure 4.8: Example 2. Model behavior in the test set. (Top) Overlapping of actual $y_{test}(t)$ and $\hat{y}_{test}(t)$. (Bottom) Scatterplot comparing the ideal and actual outputs.

It is clear that the more points used for the estimation of the BLA, the more accurate the final result of the model will be. This is not particularly surprising, however, it is interesting to note, that even when not using the maximum number of points, the results can still be very good as long as enough points are used as shown in Table 4.1. This means that a tradeoff between the number of points used and the desired accuracy is present in the method and is particularly relevant for the BLA part.

4.4.3 Noise impact and methods comparison

Once more, in order to consider the effect of noise in the results given by the method, a series of Monte Carlo simulations were run. In Fig. 4.9 the results of 100 Monte Carlo simulations are presented for examples 1 and 2. For each of the examples, different levels of noise were considered. Other than the noise, the same type of signals of Section 4.4.1 were used. In addition, in Fig. 4.10 an equivalent series of Monte Carlo simulations were run using the NARX-LSSVM approach (Suykens et al., 2002). These results allow a comparison between the proposed method and a black box approach to take place. For the NARX-LSSVM cases, the same number of realizations as used in the proposed method were averaged, thus diminishing the effective noise considerably. Note that in the proposed case this is only done for the linear block estimation.

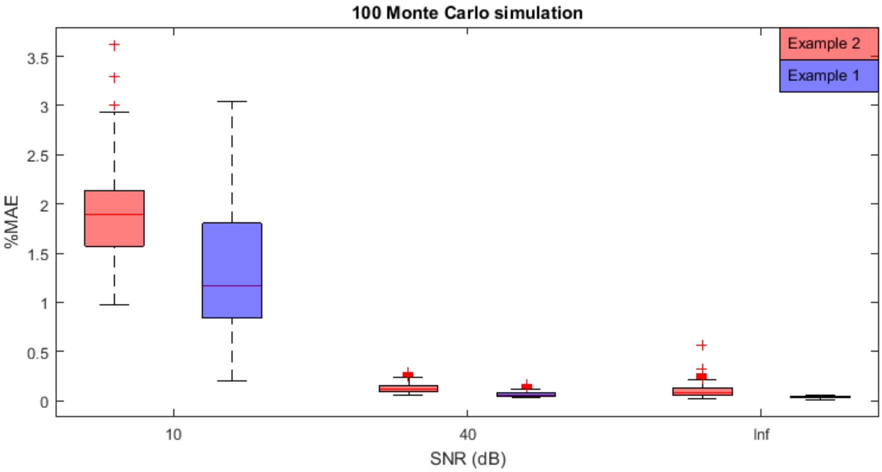


Figure 4.9: Monte Carlo simulations for the proposed method.

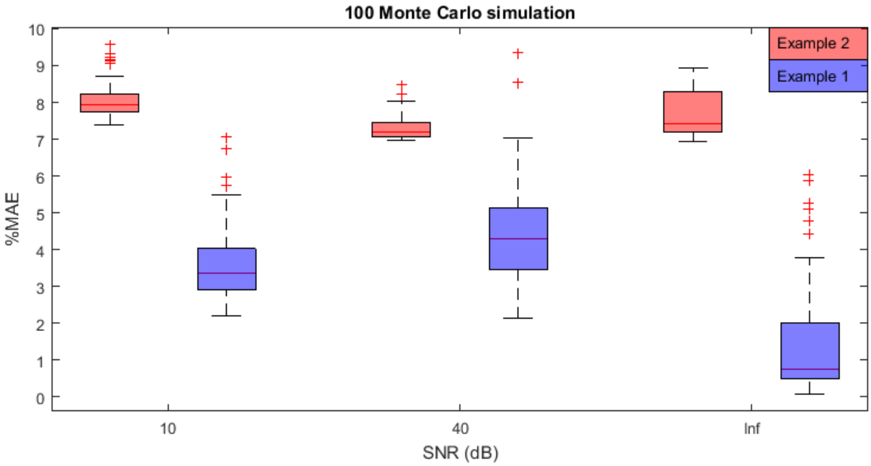


Figure 4.10: Monte Carlo simulations for NARX-LSSVM.

		SNR (dB)		
		10	40	Inf
BLA + LS-SVM	EX 1	1.1675	0.062029	0.031687
	EX 2	1.8979	0.11666	0.082519
NARX-LSSVM	EX 1	3.3492	4.2889	0.74061
	EX 2	7.944	7.1879	7.4343

Table 4.2: Summary of medians for the Monte Carlo simulations of Figs. 4.9 and 4.10

In addition, in Table 4.2, the corresponding medians of Figs. 4.9 and 4.10 are summarized for clarity. As can be seen, the proposed method outperforms the NARX-LSSVM approach at both examples with all the levels of noise used. This was to be expected as in the new method more information about the system is being included.

4.5 Conclusions

The proposed method in this chapter uses powerful techniques from two different fields. On one hand the BLA from the System Identification field and on the other, LS-SVM. When put together, these techniques are shown to be very effective for the identification of Wiener systems.

In this chapter the LS-SVM formulation was modified to include further information from the system with the help of the BLA. The results presented indicate that the method is very effective in the presence of zero mean, white Gaussian noise as long as enough samples can be measured. Indeed it can outperform powerful methods for black box modeling like NARX-LSSVM were the structure of the system is not considered.

Once all the parameters of the method (i.e. $\hat{a}_i$, $\hat{b}_j$, σ and γ) are estimated, new points can be easily evaluated. Also, the method can provide insight into the studied system as it allows to obtain models of the linear and nonlinear blocks that resemble the actual system quite accurately though in a rescaled manner. Finally, being able to draw the solution of the model from a linear system of equations is by itself an advantage over other methods like overparametrization.

An interesting extension to this method would be combining the present work with the phase coupled multisine approach proposed in J. Schoukens et al. (2014) and the Hammerstein System Identification presented in Chapter 3. Such combination would be a natural extension for the identification of Wiener-Hammerstein systems. This would be possible thanks to the capabilities of the phase coupled multisine approach to give an estimation of each of the linear blocks of such a system.

Part II

Steady State Time Response System Identification

In this part, novel methods for block oriented nonlinear system identification are offered where the common denominator is the use of the Steady State Time Response of the systems.

The methods presented in this part can be very accurate and allow a clear separation of the identified blocks.

In Chapter 5 the work presented in Castro-Garcia, Agudelo, Tiels, and Suykens (2016) is introduced. There, a straightforward estimation of the nonlinear block through the use of LS-SVM is done by making use of the behavior of SISO Hammerstein systems in steady state. Using the estimated nonlinear block, the intermediate variable is calculated. Finally using the latter and the known output, the linear block can be estimated.

The approach of Chapter 5 is extended in Chapter 6 where the MIMO case is considered. The method presented consists of two stages. In the first stage LS-SVM is used to model the nonlinear block of the Hammerstein system from its steady-state response. In the second stage, the intermediate variables are computed by using the previously estimated nonlinear block. Then, the linear block is estimated from the latter and the known outputs by using subspace identification methods. The presented methodology is very flexible concerning the class of problems it can handle and no previous knowledge about the underlying non-linearities is required except for very mild assumptions. It is particularly effective when dealing with hard to model nonlinearities where other methods often fail. Also, it can handle different numbers of inputs/outputs and performs well in the presence of white Gaussian noise. This chapter is based on the work presented in Castro-Garcia, Agudelo, and Suykens (2017c).

Finallly in Chapter 7 we propose a new methodology for identifying Wiener systems using the data acquired from two separate experiments. In the first experiment, we feed the system with a sinusoid at a prescribed frequency and use the steady state response of the system to estimate the static nonlinearity. In the second experiment, the estimated nonlinearity is used to identify a model of the linear block feeding the system with a persistently exciting input. We discuss both parametric and nonparametric approaches to estimate the static nonlinearity. In the parametric case, we show that modeling the static nonlinearity as a polynomial results into a fast least-squares based estimation procedure. In the nonparametric case, LS-SVM are employed to obtain a flexible model. This chapter is based on the works presented in Bottegal, Castro-Garcia, and Suykens (2017a, 2017b).

Chapter 5

Hammerstein System Identification using LS-SVM and Steady State Time Response

5.1 Introduction

In this chapter the q-notation will be used. The operator q is a time shift operator of the form $q^{-1}x(t) = x(t-1)$.

The idea in this chapter is to use Least Squares Support Vector Machines (LS-SVM) Suykens et al. (2002) while making use of the characteristic behavior of Hammerstein systems under steady state. The resulting methodology turns out to be easily implementable while giving good results. Also, it allows to separate the identification of the linear and nonlinear parts.

Although previous works in the system identification literature have used LS-SVM (e.g. see Falck et al. (2012, 2009); Goethals et al. (2005)), none of them have attempted a straightforward calculation of the nonlinear block using LS-SVM.

This chapter is based on:
Castro-Garcia, R., Agudelo, O. M., Tiels, K., Suykens, J. A. (2016). *Hammerstein system identification using LS-SVM and steady state time response*. In proceedings of the 15th European Control Conference (pp. 1063 – 1068).

The proposed method is based on applying a multilevel input signal in which the duration of the steps is longer than the settling time of the system. It uses a forward approach as defined in Sun et al. (1999) where the nonlinear block is identified first, and the linear block is modeled afterwards. More precisely, the method consists of the following steps:

- The system's settling time is estimated through the application of a step signal.
- A multilevel input signal is created based on the calculated settling time.
- An LS-SVM model is trained using the levels of the multilevel signal as inputs and their corresponding output values in steady state as outputs.
- An additional experiment is carried out in order to identify the linear block. Here the applied input is evaluated using the obtained nonlinearity in order to estimate the intermediate variable. With the intermediate variable and the known output, the linear block is estimated through least squares.

A somewhat similar approach was proposed in Ikhouane and Giri (2014). However, there it is assumed that the nonlinearity is a linear combination of known functions and that it is locally invertible. In this work, those assumptions are not necessary. Additionally, in this chapter a way for identifying Hammerstein systems for which the linear block is a high pass filter is offered. This is not possible with the method offered in Ikhouane and Giri (2014).

The proposed method provides an easy way to directly use standard LS-SVM for the identification of Hammerstein systems while bearing in mind the structure of such systems. It allows to estimate the nonlinear block in a straightforward manner independently of the linear block and does not require any particularly complex set of inputs-outputs. This is important as it implies that the method can be applied to a wide set of problems. Also, given the way it works, it can give very good approximations to the intermediate variable (up to a scaling factor) even in the presence of heavy white Gaussian zero mean noise.

The chapter is organized as follows: In Section 5.2, the proposed methodology is presented. Section 5.3 illustrates the results found when applying the described methodology on two simulation examples. Finally, in Section 5.4, the conclusions are presented.

5.2 Proposed Method

In this method, the first step is to construct a data set where the input $u_1(t)$ is a multilevel signal in which each step lasts a constant amount of time T_C defined as:

$$T_C = T_S + \Delta_T, \tag{5.1}$$

where T_S is the settling time of the system and Δ_T is an arbitrary additional time. This way of constructing $u_1(t)$ guarantees that during each step of the input signal some samples will be taken after the system has reached steady state (i.e. those taken during Δ_T after T_S). The input signal $u_1(t)$ can then be described as:

$$u_1(t) = r_k, \text{ for } kT_C \leq t < (k+1)T_C. \tag{5.2}$$

For each of the steps $k \in \mathbb{N}$, $u_1(t)$ has a constant value r_k.

The settling time of the system T_S is estimated by applying a step signal to the system and determining the time it takes for the corresponding output to stay within a certain range.

It is assumed that the linear block is stable (i.e. all the poles are inside the unit circle). Also, it is assumed for now that the step response of the system does not tend to zero as time tends to infinity, that is:

$$\lim_{t\to\infty} y(t) \neq 0, \tag{5.3}$$

for

$$x(t) = \left\{ \begin{array}{l} 0,\ t < 0 \\ r,\ 0 \leq t < \infty. \end{array} \right. \quad \text{with } r \neq 0 \tag{5.4}$$

In Section 5.3.5 a way for overcoming this limitation is presented.

The samples of the output $y_1(t)$ taken during $kT_C + T_S \leq t < (k+1)T_C$ are averaged for each k in order to minimize the effect of the measurement noise during each step.

In Fig. 5.1 an excerpt of a training signal is shown to illustrate the samples taken after the settling time at each step of the signal. The red boxes indicate the values of the output signal that are averaged for each step.

Let us define $\tilde{u}_1(k) = r_k$, a signal containing the amplitude level of each step of the input signal. Also, let us define $\tilde{y}(k)$, a signal containing the output averages corresponding to the inputs during $kT_C + T_S \leq t < (k+1)T_C$. Using $\tilde{u}(k)$ as input and $\tilde{y}(k)$ as output, an LS-SVM model can be trained. For the example shown in Fig. 5.1, the corresponding extracted values $\tilde{u}(k)$ and $\tilde{y}(k)$ are presented in Fig. 5.2.

In this chapter, LS-SVM is used under a 10-fold cross validation setting to obtain the estimation of the nonlinear block. Once this is done, another experiment is carried

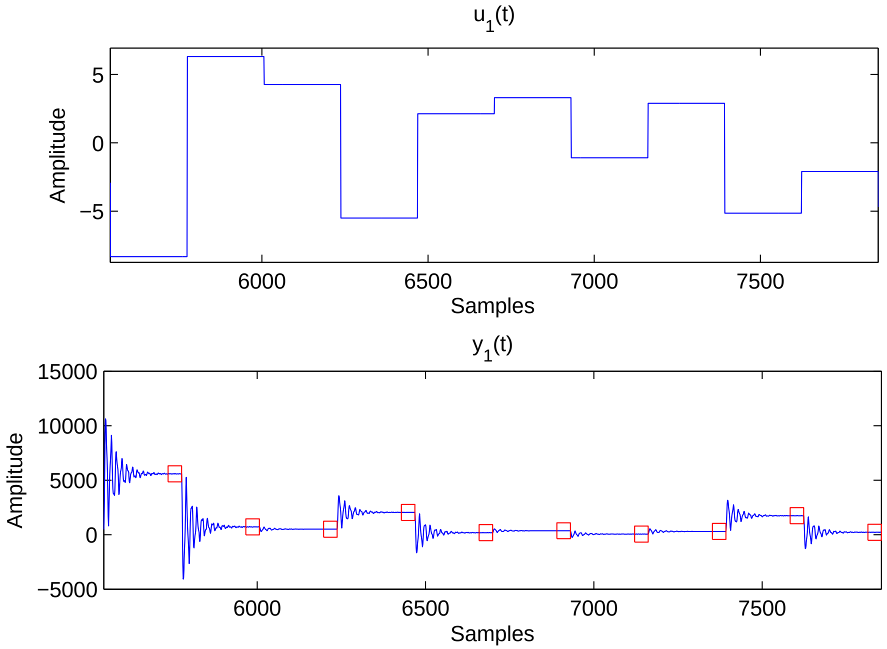


Figure 5.1: Example of a training signal. (Top) Input signal $u_1(t)$. (Bottom) Output signal $y_1(t)$.

out, where a new input signal $u_2(t)$ is generated and its corresponding output $y_2(t)$ is obtained. This input signal is then evaluated using the estimated nonlinearity to obtain an approximation to the intermediate variable $x_2(t)$ (i.e. $\hat{x}_2(t)$).

The linear block is a discrete-time rational transfer function of the form

$$\hat{y}(t) = \sum_{j=0}^{m} b_j x(t-j) - \sum_{i=1}^{n} a_i y(t-i), \tag{5.5}$$

and so, $y_2(t) = \sum_{j=0}^{m} \hat{b}_j \hat{x}_2(t-j) - \sum_{i=1}^{n} \hat{a}_i y_2(t-i)$. The coefficients $\hat{b}_j$ and $\hat{a}_i$ are estimated here using standard least squares to find an approximation of the linear block. This is done using $\hat{x}_2(t)$ and the known output $y_2(t)$. During this step, several orders for the numerator and denominator are tried out.

In Fig. 5.3, a simplified summary of the method is presented.

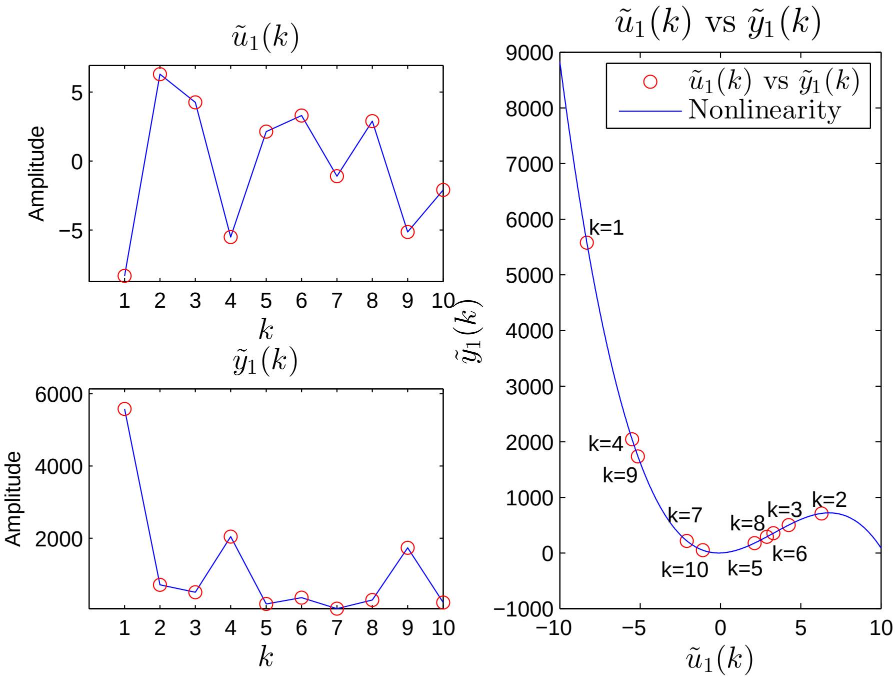


Figure 5.2: Corresponding training points for the example of Fig. 5.1. (Top-Left) r_k values. (Bottom-Left) Averaged $y_1(t)$ values. (Right) $\tilde{u}(k)$ vs $\tilde{y}(k)$ and the rescaled nonlinearity.

5.3 Results

5.3.1 Example

The proposed methodology was applied to a system in the discrete time domain. In order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D. The system was generated through a nonlinear block:

$$x(t) = \frac{u(t)^2 \sin(\pi u(t))}{\pi u(t)} \tag{5.6}$$

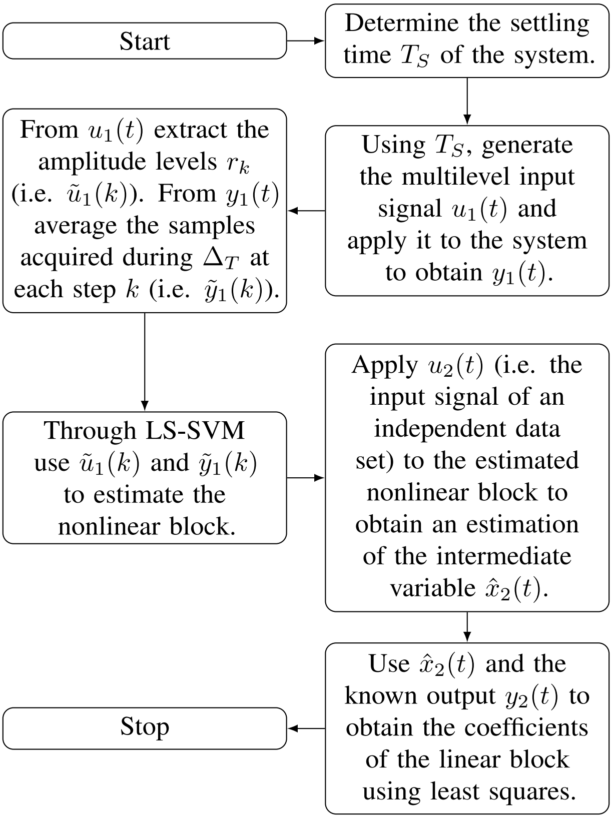

Start
Determine the settling time T_S of the system.
Using T_S, generate the multilevel input signal $u_1(t)$ and apply it to the system to obtain $y_1(t)$.
From $u_1(t)$ extract the amplitude levels r_k (i.e. $\tilde{u}_1(k)$). From $y_1(t)$ average the samples acquired during Δ_T at each step k (i.e. $\tilde{y}_1(k)$).
Through LS-SVM use $\tilde{u}_1(k)$ and $\tilde{y}_1(k)$ to estimate the nonlinear block.
Apply $u_2(t)$ (i.e. the input signal of an independent data set) to the estimated nonlinear block to obtain an estimation of the intermediate variable $\hat{x}_2(t)$.
Use $\hat{x}_2(t)$ and the known output $y_2(t)$ to obtain the coefficients of the linear block using least squares.
Stop

Figure 5.3: Summary of the method.

and a linear block:

$$y(t) = \frac{0.1129q^4 - 0.2128q^3 + 0.283q^2 - 0.2128q + 0.1129}{q^4 - 2.485q^3 + 2.528q^2 - 1.184q + 0.2245} x(t). \tag{5.7}$$

The system is shown in Fig. 5.4.

5.3.2 Signals description

To construct $u_1(t)$, the settling time T_S was established first by exciting the system with a step of amplitude 10. In this example $T_S = 191$ samples. Afterward, the signal

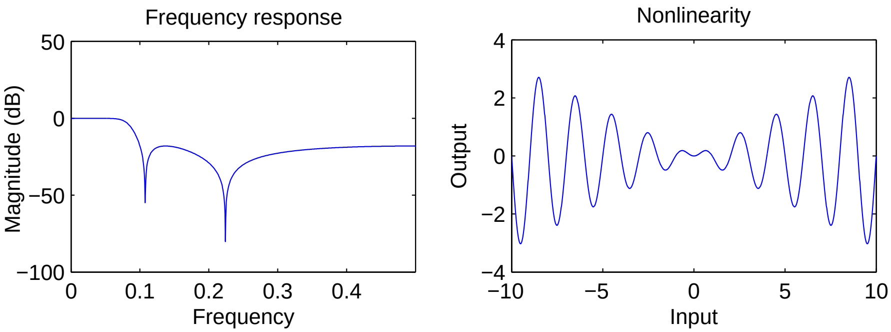


Figure 5.4: (Left) Linear block representation in the frequency domain (normalized frequency). (Right) Nonlinear block representation in the time domain.

was constructed by adding 40 extra samples at each step to those required to achieve steady state (i.e. Δ_T). The amplitudes of the steps in this signal (i.e. r_k) were randomly drawn from a uniform distribution ranging between -10 and 10.

From the resulting $y_1(t)$ the values corresponding to the output of the last Δ_T samples at each step were retrieved and averaged (i.e. $\tilde{y}(k)$). In order to estimate the nonlinear block, 500 input-output pairs were used.

In Fig. 5.5 the resulting nonlinear block of the example is compared with the actual one for a run with a Signal to Noise Ratio (SNR) of 40dB. Note that a rescaling constant is present there. If both, the linear and nonlinear, blocks are considered, there will be a gain factor of the combined blocks. However, this gain could be distributed in any way between the two blocks Boyd and Chua (1983). The actual difference in scaling has no effect on the input-output behavior of the Hammerstein system (i.e. any pair of $\{f(u(t))/\eta,\ \eta G(q)\}$ with $\eta \neq 0$ would yield identical input and output measurements). Up to this scaling factor, it is clear that the estimated nonlinear block is a good representation of the actual one.

To estimate the linear block, a new data set of 5000 points was generated. $u_2(t)$, the input to generate this data set, is a multilevel signal where each step has a duration $T_C = 10$ samples. The amplitudes at each level were drawn from a uniform distribution ranging between -10 and 10.

Using $u_2(t)$ and the estimated nonlinear block, an estimation of the intermediate variable $\hat{x}_2(t)$ is calculated. Using $\hat{x}_2(t)$ and the known output $y_2(t)$, the linear block is estimated through least squares. Orders ranging between 1 and 10 were tried out for numerator and denominator. Note that in order to fulfill the made assumptions, given a

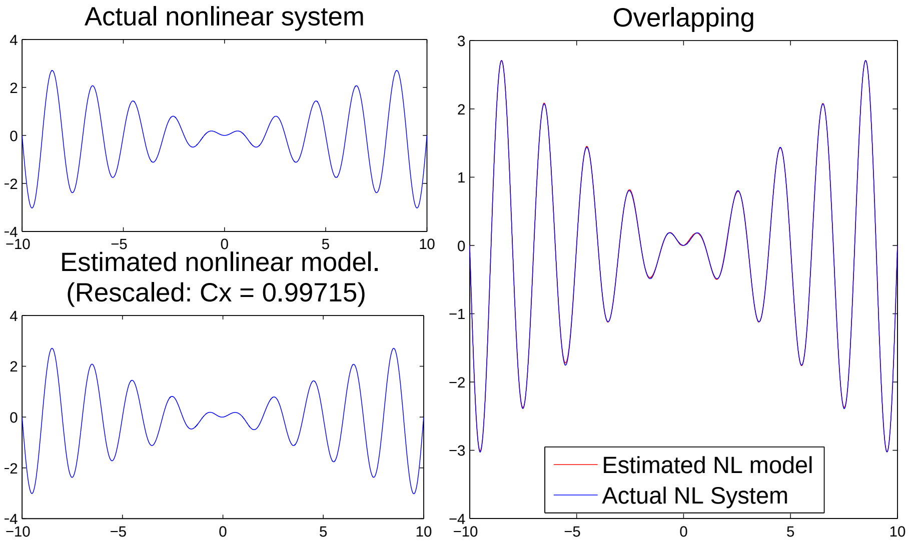


Figure 5.5: In all the plots, the vertical axes represent the output value for the corresponding values in the horizontal axes..

linear block defined as in (5.5), only cases where $m \leq n$ can be considered.

Finally, the system was tested in a third data set. The input for generating this set, $u_{test}(t)$, is a multilevel signal where each step has a duration $T_C = 10$ samples. The amplitudes at each level were drawn from a uniform distribution ranging between -10 and 10. This data set consists of 5000 points.

In Fig. 5.6 the estimated output is compared with the actual one for the same run used in Fig. 5.5.

Note that white Gaussian noise with zero mean was applied to the output of each data set. In Section 5.3.3 the effect of noise in the method is explained.

5.3.3 Noise effect analysis

In order to evaluate how the noise affects the performance of the proposed method, 100 Monte Carlo simulations were carried out for each of four different SNRs varying between 10dB and 80dB.

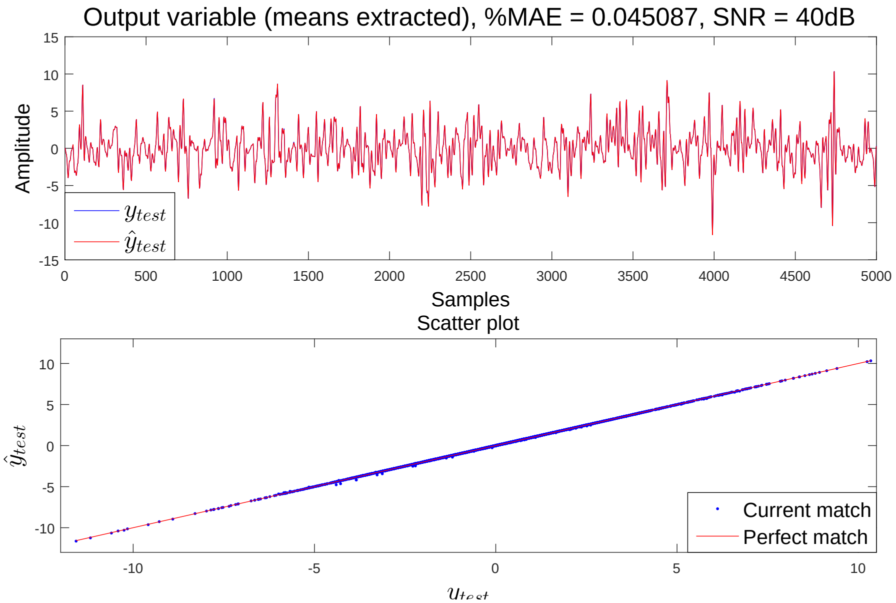


Figure 5.6: (Top) Overlapping of the actual and estimated output variables. (Bottom) Scatter plot illustrating the behavior of the overlapped plots.

In Fig. 5.7 the results of the Monte Carlo simulations are presented. As can be seen, the performance of the proposed method dramatically changes as the level of the noise varies.

It is important to highlight that the impact of noise can be further reduced if more points are considered in the data set employed for estimating the nonlinear block. To illustrate this, in Fig 5.8 it is shown how the performance of the method changes for the example when using a SNR of 10dB.

5.3.4 Methods comparison

The proposed method was compared with:

- A NARX LS-SVM (Suykens et al., 2002) with 10 lags of input and 10 lags of output.

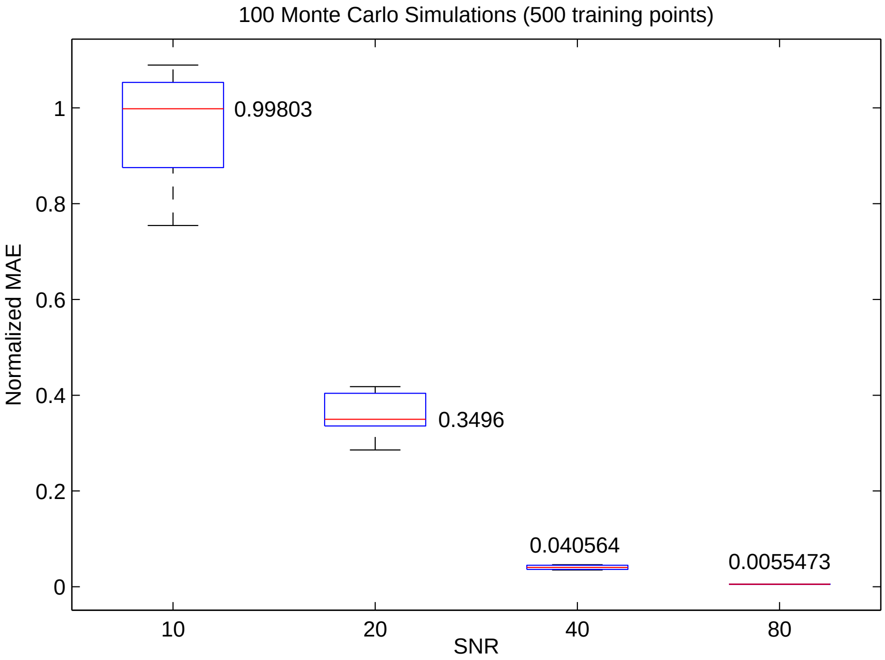


Figure 5.7: Evolution of the normalized MAE of the output of the model as the SNR changes. The corresponding median values appear next to each box.

- The Hammerstein and Wiener Identification procedure (in this chapter denoted by WHIP) presented in M. Schoukens (2015).
- The iterative method (in this chapter denoted by IM) presented in Bai and Li (2010).

The proposed method was implemented using a RBF kernel for the LS-SVM part. This kernel requires the tuning of a kernel parameter σ and a regularization parameter γ (Suykens et al., 2002).

For the IM method a Gaussian noise input was used. This signal had a standard deviation as large as the standard deviation of the concatenation of the input signals $u_1(t)$ and $u_2(t)$ as described in Section 5.3.2. The models were estimated using 115500 samples, while 5000 samples were used to look for the best model order (i.e. scan over orders 2, 3, 4, 5, and 6). To model the nonlinearity a piecewise linear function with

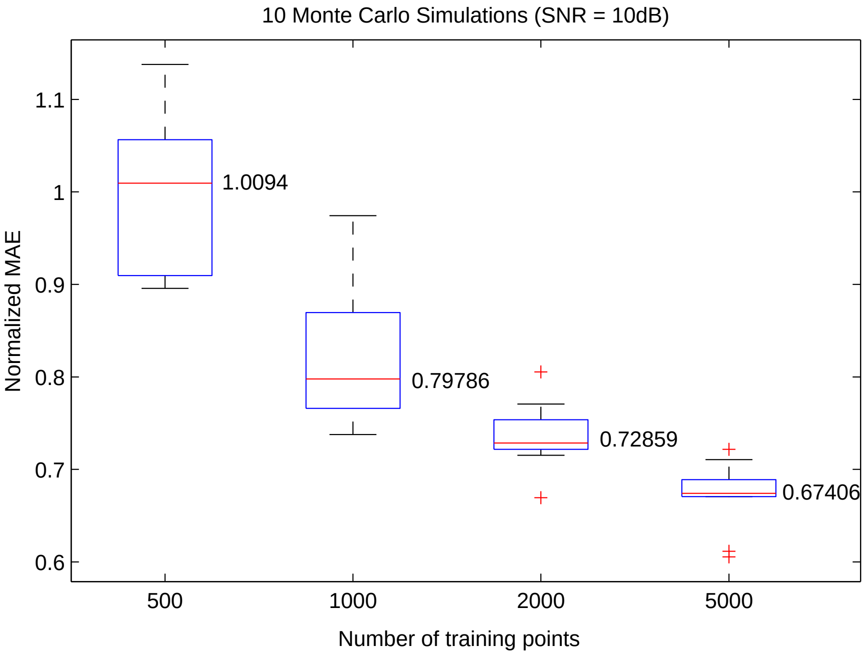


Figure 5.8: Evolution of the normalized MAE of the output of the model as the number of training points changes. The corresponding median values appear next to each box.

50 breakpoints was used. It is important to note that the choice for the input signal of the IM method is such that as many samples and as much total energy is used for the identification of the system as for the proposed method.

For the WHIP method a random-phase multisine was employed. Again, this signal had the same standard deviation as the concatenation of signals $u_1(t)$ and $u_2(t)$. Seven phase realizations and 2 periods (plus an additional period to reduce the effect of transients) of the multisine with 5000 samples per period were used to estimate the models. One period (no transient removal) of an additional realization was used to look for the best model order with the same order scanning used for the IM method.

In Table 5.1 the results of the comparison in Normalized MAE form are presented. Each of the presented results corresponds to an average over 10 runs.

The results indicate that the proposed method obtains better results as the noise is reduced. For the NARX LS-SVM the results seem to stay almost the same as the noise

Table 5.1: Results comparison in Normalized MAE on test data.

	SNR (dB)	
	10	**20**
Proposed method	1.5259	0.6925
NARX LS-SVM	9.9266	9.9314
IM	8.8621	9.3660
WHIP	5.6204	8.3097

is increased. For WHIP and IM, the results are better when the noise is increased. This result is explained by the presence of outliers in the results, which indicates that these methods are sensitive to local minima.

The IM and WHIP methods assume that the nonlinearity can be represented in a basis function expansion form with known basis functions. Note that the nonlinearity in (5.6) is hard to model by a polynomial of reasonable degree and in consequence, piecewise linear basis functions were used. Since a finite number of breakpoints is used, the true nonlinearity is not in the model class. This can be an explanation for the poor results of the last two methods in the example.

As can be seen, the proposed method performs very well in the example. This behavior suggests that it is robust against the amount of noise used.

5.3.5 High pass filter case

The proposed methodology gives good results in the established framework. However, as it is presented, the method is unable to deal with situations where the assumption introduced in Eqs. (5.3) and (5.4) is violated. A clear illustration of this occurs when the linear block is a high pass filter. In this particular situation:

$$\lim_{t\to\infty} y(t) = 0, \tag{5.8}$$

for

$$x(t) = \begin{cases} 0, \; t < 0 \\ r, \; 0 \leq t < \infty. \end{cases} \quad \text{with } r \neq 0. \tag{5.9}$$

In this case, for training the LS-SVM model, the corresponding output points would always be zero or very close to zero:

$$\tilde{y}(k) = 0 \, \forall k. \tag{5.10}$$

In order for the method to be able to work in these situations, the addition of one or several integrators to the output signal is proposed, this is represented in Fig. 5.9. Note

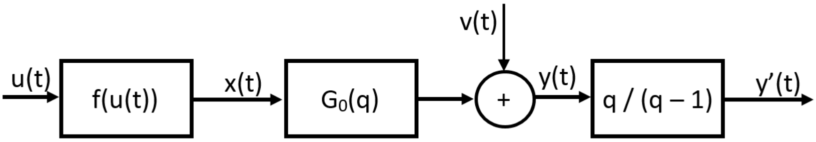


Figure 5.9: Hammerstein system with an added integrator at the output for estimation of the nonlinear block.

that this has to be done only in the first stage of the method, that is, for the estimation of the nonlinear block. The number of integrators required depends directly on the linear block. However, it can be easily established through direct observation. If more integrators than needed are added, the system will become unstable.

To illustrate the high pass filter case, an example is presented where the nonlinear block has the form

$$x(t) = u(t) + 5u(t)^2 - \frac{u(t)^3}{2} \tag{5.11}$$

and the linear block is given by:

$$y(t) = \frac{q^2 - 1.8q + 0.8}{q^2 - 1.5q + 0.7225} x(t). \tag{5.12}$$

This system is illustrated in Fig. 5.10. In this example, the signals used are very similar to those described in Section 5.3.2, however, 100 $\{\tilde{u}(k), \tilde{y}(k)\}$ pairs were used instead of 500. Also, the second data set (i.e. $\{u_2(t), y_2(t)\}$) and the test set consisted of 1000 samples.

Once the linear block is estimated as explained in Section 5.2, the model of the system is tested with an independent data set. The resulting output variable behavior is presented in Fig. 5.11.

In Fig. 5.12 the results of a Monte Carlo simulation of 10 runs for different levels of noise is shown. It shows how the normalized MAE evolves as the level of noise changes in the example represented by Eqs. (5.11) and (5.12).

Note that this approach can be sensitive to cases with zeros very close to 1 but not exactly at 1 in the unit circle. In those cases, using the proposed method with both the non-integrated or the integrated output might yield unsatisfactory results.

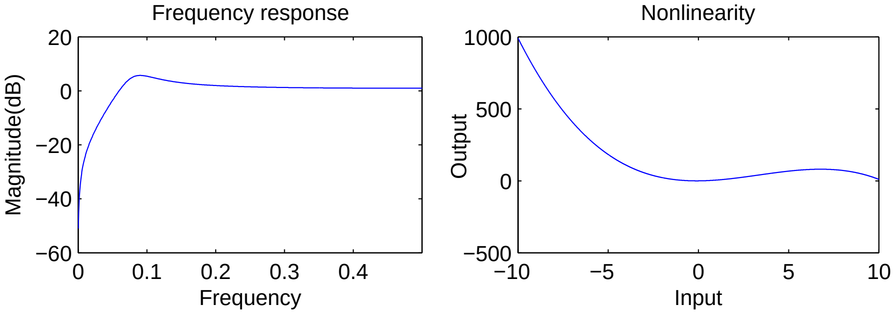


Figure 5.10: High pass filter example: (Left) Linear block representation in the frequency domain (normalized frequency). (Right) Nonlinear block representation in the time domain.

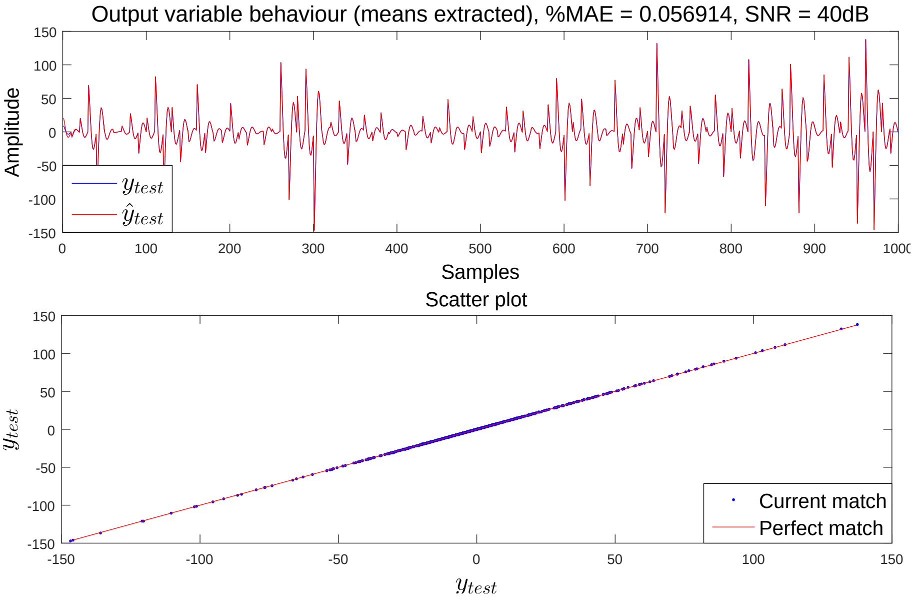


Figure 5.11: High pass filter example: (Top) Overlapping of the actual and estimated output variables. (Bottom) Scatter plot illustrating the behavior of the overlapped plots.

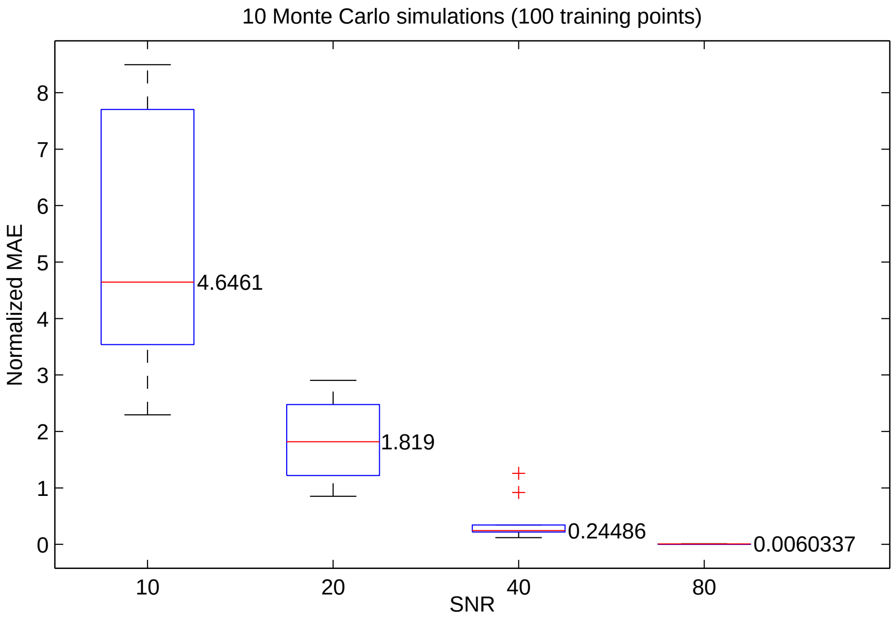


Figure 5.12: High pass filter example: Normalized MAE for different levels of noise. The corresponding median values appear next to each box.

5.4 Conclusions

The method presented in this chapter offers a simple way for accurate Hammerstein system identification. This is done mainly by making use of the behavior of the system in steady state. In this work, this was done through LS-SVM which allows a good generalization when using different model classes.

The main strength of the proposed method lies in the identification of the nonlinear block of Hammerstein systems. The presented results indicate that the method is very effective in the presence of zero mean, white Gaussian noise.

Once the nonlinear model is learned, it can be easily applied. It is shown that even with a small amount of training points, the results are already quite accurate. In practice, this means that the calculation of the model can also be done very quickly. It is also possible to improve the performance of the method by using more training points for modeling the nonlinearity.

The estimated nonlinear model is very close to the original one (up to a scaling factor). This allows insight into the behavior of the studied system as it is possible to visualize the way the nonlinear block will respond to the inputs. Naturally, this allows as well a good estimation of the intermediate variable.

The way $u_1(t)$ is constructed is quite simple and given its shape, it allows the application of the method in many fields. However, a possible drawback of the methodology lies in the fact that depending on the evaluated system, constructing the initial input signal $u_1(t)$ could require a long time. Nevertheless, the central idea of this work can be used in the identification of Wiener and Wiener-Hammerstein systems as the working concepts would be basically the same. Though not as straightforward as in the Hammerstein case, full models of these structures could be estimated after the nonlinearity is modeled. Also, more complex cases like MIMO Hammerstein (see Chapter 6), Wiener (see Chapter 7) and Wiener-Hammerstein can also be considered though they will not be as easily adapted.

Chapter 6

MIMO Hammerstein System Identification using LS-SVM and Steady State Time Response

6.1 Introduction

Most of the works regarding Hammerstein System Identification are focused on the Single-Input Single-Output (SISO) case while the Multiple-Input Multiple-Output (MIMO) case has received much less attention. Methods dealing with the MIMO case include for instance: In Gomez and Baeyens (2004) basis functions are used to represent both the linear and nonlinear parts of Hammerstein models; in Jeng and Huang (2008), through the use of specially designed signals, the impulse response of the system is estimated and through least squares the intermediate variables are computed. Using this approximation and the known input, a mapping of the nonlinearity is done through the fitting of a polynomial; an overparametrization approach is proposed in Goethals et al. (2005) in combination with a reformulated version of LS-SVM, although the MIMO case is not actually tested. Other methods for MIMO Hammerstein system

This chapter is based on:
Castro-Garcia, R., Agudelo, O. M., Suykens, J. A. K. (2017c). *MIMO Hammerstein System Identification using LS-SVM and Steady State Time Response*. Accepted for publication in the proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI 2017). (Internal Report 17-23, ESAT-SISTA, KU Leuven. Leuven, Belgium).

identification can be found in Lee et al. (2004); Verhaegen and Westwick (1996) and Al-Duwaish and Karim (1997).

The method proposed in this chapter consists of two stages. In the first one multilevel input signals with a step duration longer than the system's settling time are applied to the process. Next, the levels of the input signals are paired with the steady-state values of each of the outputs. The scalar functions that are part of the nonlinear block (here, it is assumed that the number of intermediate variables is equal to the number of inputs) are approximated from the previously found input-output mappings using LS-SVM. In the second stage, an additional experiment is carried out in order to identify the linear part. Here the input signals are evaluated in the obtained nonlinear scalar functions in order to estimate the intermediate variables. With these estimations and the known outputs of the system, the linear block is identified using subspace methods (i.e., N4SID Van Overschee and De Moor (1996)).

Due to the use of LS-SVM to model the nonlinear part, the proposed method is very flexible regarding the class of systems that can be modeled. For instance, whereas the work in Lee et al. (2004) is applicable only to the case where the nonlinearities are in terms of a polynomial, or in Gomez and Baeyens (2004) specific basis functions have to be chosen beforehand, the methodology presented in this work is free of these limitations thanks to the good generalization properties of LS-SVM.

The proposed method was tested in two examples through several Monte Carlo simulations. It will be illustrated how the measurement noise (white Gaussian noise with zero mean) affects its behavior and also how its accuracy compares with other state of the art methodologies.

This chapter is organized as follows. In Section 6.2, the proposed method is presented. Section 6.3 shows the results found when applying the described methodology on two simulation examples. Finally, in Section 6.4, the conclusions are exposed.

6.2 Proposed Method

The proposed methodology is an extension of the work in Chapter 5, where a method for SISO Hammerstein identification using steady state information is offered.

In this chapter it is assumed that the system will have as many intermediate variables as inputs. Additionally, for a system with p inputs it is assumed that $f_i(\mathbf{0}_p) = 0$ for $i = 1, \ldots, p$.

For the sake of clarity and without loss of generality, let us consider a system with 2 inputs $u_1(t)$ and $u_2(t)$, and 2 outputs $y_1(t)$ and $y_2(t)$, as the one shown in Fig. 6.1. In order to estimate $f_1(\cdot)$ and $f_2(\cdot)$, we first excite the system with multilevel signals

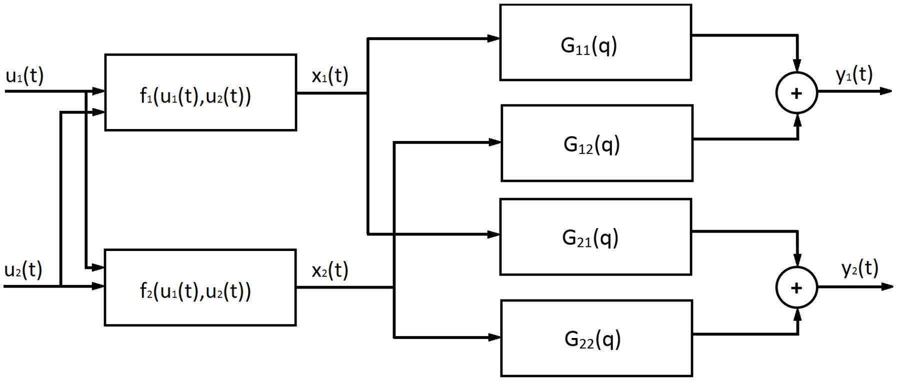


Figure 6.1: MIMO Hammerstein system with 2 inputs and 2 outputs.

$u_1(t)$ and $u_2(t)$ defined as follows:

$$\begin{aligned} u_1(t) &= r_k, \quad \text{for} \quad kT_{C,1} \leq t < (k+1)T_{C,1} \\ u_2(t) &= w_i, \quad \text{for} \quad iT_{C,2} \leq t < (i+1)T_{C,2}. \end{aligned} \tag{6.1}$$

This means that for each of the steps k and $i \in \mathbb{N}$, $u_1(t)$ has a constant value r_k and $u_2(t)$ has a constant value w_i. $T_{C,1}$ and $T_{C,2}$ are the amount of time that $u_1(t)$ and $u_2(t)$ are kept constant and are defined as follows:

$$\begin{aligned} T_{C,1} &= T_S + \Delta_T \\ T_{C,2} &= N_s T_{C,1}. \end{aligned} \tag{6.2}$$

Here, T_S is the settling time of the system, Δ_T is an arbitrary additional time and N_s is the number of levels of $u_1(t)$ to be tried out per level of $u_2(t)$. This way of constructing $u_1(t)$ and $u_2(t)$ allows a proper sweep of the possible combinations of the inputs. Figure 6.2 shows an example of how these signals look like. Δ_T guarantees that during each step of $u_1(t)$ some samples will be taken after the system has reached steady state (i.e. those taken during Δ_T after T_S).

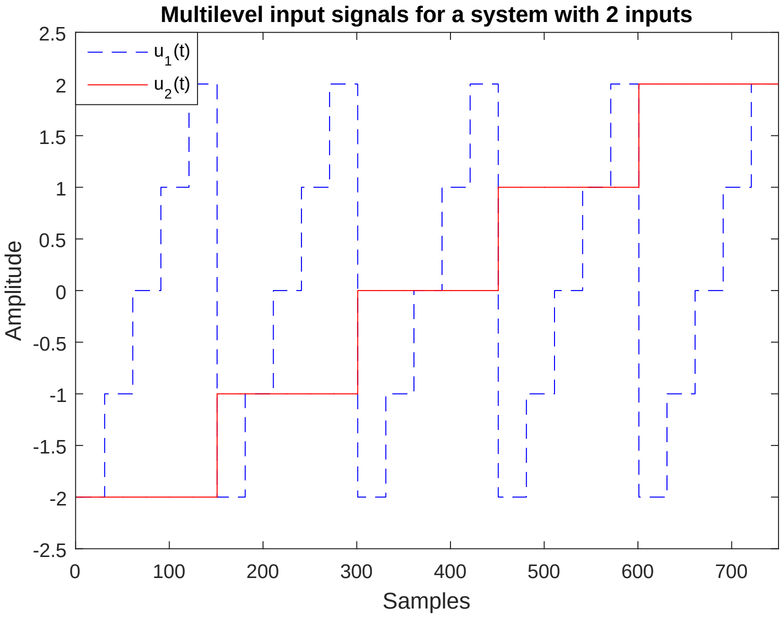


Figure 6.2: Multilevel input signals for a system with 2 inputs.

Now, let us define the vectors $\tilde{\boldsymbol{u}}_1 \in \mathbb{R}^{N_1 N_2}$ and $\tilde{\boldsymbol{u}}_2 \in \mathbb{R}^{N_1 N_2}$ containing the amplitude levels of the input signals

$$\tilde{\boldsymbol{u}}_1 = \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N_1-1} \\ r_0 \\ r_1 \\ \vdots \\ r_{N_1-1} \\ \vdots \\ r_0 \\ \vdots \\ r_{N_1-1} \end{bmatrix}, \; \tilde{\boldsymbol{u}}_2 = \begin{bmatrix} w_0 \\ w_0 \\ \vdots \\ w_0 \\ w_1 \\ w_1 \\ \vdots \\ w_1 \\ \vdots \\ w_{N_2-1} \\ \vdots \\ w_{N_2-1} \end{bmatrix}, \tag{6.3}$$

where N_1 and N_2 are the number of levels of $u_1(t)$ and $u_2(t)$ respectively. Also, let us define the vectors $\tilde{\boldsymbol{y}}_1 \in \mathbb{R}^{N_1 N_2}$, and $\tilde{\boldsymbol{y}}_2 \in \mathbb{R}^{N_1 N_2}$ where the samples of the outputs $y_1(t)$ and $y_2(t)$ taken during $kT_{C,1} + T_S < t < (k+1)T_{C,1}$ are averaged for each k in order to minimize the effect of the measurement noise during each step

$$\begin{aligned} \tilde{\boldsymbol{y}}_{1,k} &= \frac{1}{N_{\Delta_T}} \sum_{t=kT_{C,1}+T_S}^{kT_{C,1}+T_S+\Delta_T} y_1(t), \\ \tilde{\boldsymbol{y}}_{2,k} &= \frac{1}{N_{\Delta_T}} \sum_{t=kT_{C,1}+T_S}^{kT_{C,1}+T_S+\Delta_T} y_2(t), \end{aligned} \tag{6.4}$$

for $k = 1, \ldots, N_1 N_2$ and with N_{Δ_T} the number of samples taken during Δ_T.

Using $\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{u}}_2$ as inputs and $\tilde{\boldsymbol{y}}_1$ as an output, an LS-SVM model can be trained to approximate the first nonlinearity $\hat{f}_1(\cdot)$ of the system. In a similar fashion, using $\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{u}}_2$ as inputs and $\tilde{\boldsymbol{y}}_2$ as an output, another LS-SVM model can be trained to approximate the second nonlinearity of the system $\hat{f}_2(\cdot)$ (See Fig. 6.3).

Notice that

$$\begin{aligned} \hat{f}_1(\cdot) &= k_{11} f_1(\cdot) + k_{12} f_2(\cdot) \\ \hat{f}_2(\cdot) &= k_{21} f_1(\cdot) + k_{22} f_2(\cdot), \end{aligned} \tag{6.5}$$

where k_{11}, k_{12}, k_{21} and k_{22} are the steady state gains of $G_{11}(q)$, $G_{12}(q)$, $G_{21}(q)$ and $G_{22}(q)$ respectively.

With models corresponding to the nonlinear part available (i.e. $\hat{f}_1(\cdot)$ and $\hat{f}_2(\cdot)$), the second stage of the method can take place. On this stage, an independent experiment is performed where the system is fed with inputs $u_{l,1}(t)$ and $u_{l,2}(t)$ and the corresponding outputs $y_{l,1}(t)$ and $y_{l,2}(t)$ are measured. Then, the intermediate variables $\hat{x}_{l,1}(t) = \hat{f}_1(u_{l,1}(t), u_{l,2}(t))$ and $\hat{x}_{l,2}(t) = \hat{f}_2(u_{l,1}(t), u_{l,2}(t))$ are computed. With $\hat{x}_{l,1}(t)$ and $\hat{x}_{l,2}(t)$ and the known outputs $y_{l,1}(t)$ and $y_{l,2}(t)$, subspace methods can be used to obtain a model of the linear block. Note that the number of intermediate variables estimated in this way will be equal to the number of outputs. For non-square systems this implies that the estimated model will have a different internal structure than the actual one.

In Fig. 6.4, a summary of the method is presented.

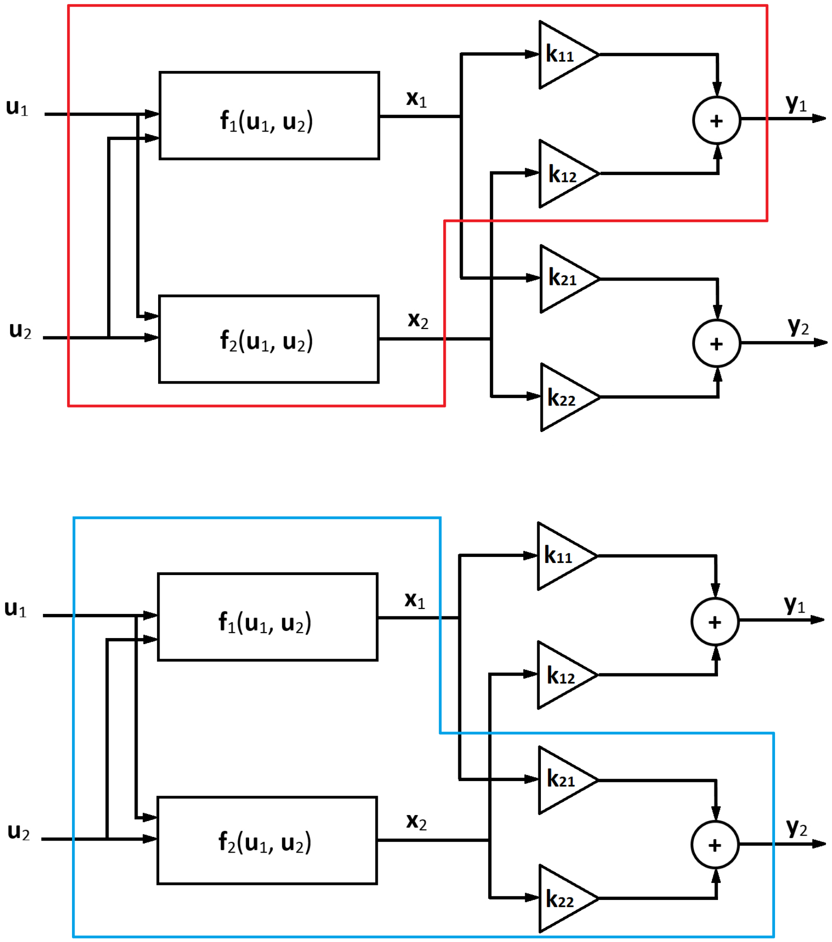


Figure 6.3: Modeling of the nonlinear block of a system with two inputs and two outputs. (Red) Nonlinearity corresponding to the output $y_1(t)$. (Blue) Nonlinearity corresponding to the output $y_2(t)$.

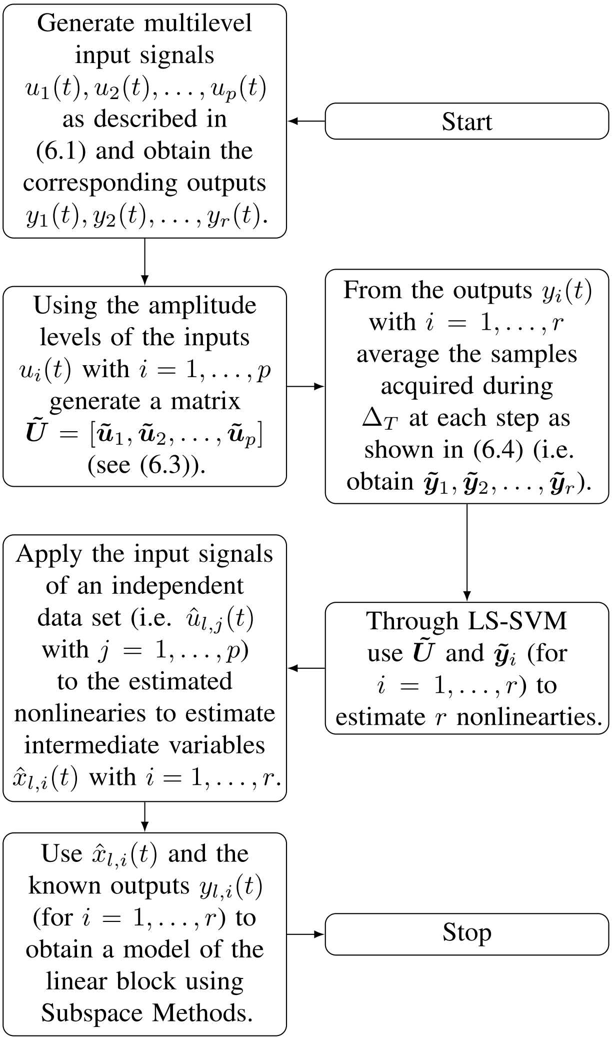

Start
Generate multilevel input signals $u_1(t), u_2(t), \ldots, u_p(t)$ as described in (6.1) and obtain the corresponding outputs $y_1(t), y_2(t), \ldots, y_r(t)$.
Using the amplitude levels of the inputs $u_i(t)$ with $i = 1, \ldots, p$ generate a matrix $\tilde{\boldsymbol{U}} = [\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \ldots, \tilde{\boldsymbol{u}}_p]$ (see (6.3)).
From the outputs $y_i(t)$ with $i = 1, \ldots, r$ average the samples acquired during Δ_T at each step as shown in (6.4) (i.e. obtain $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots, \tilde{\boldsymbol{y}}_r$).
Through LS-SVM use $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{y}}_i$ (for $i = 1, \ldots, r$) to estimate r nonlinearties.
Apply the input signals of an independent data set (i.e. $\hat{u}_{l,j}(t)$ with $j = 1, \ldots, p$) to the estimated nonlinearies to estimate intermediate variables $\hat{x}_{l,i}(t)$ with $i = 1, \ldots, r$.
Use $\hat{x}_{l,i}(t)$ and the known outputs $y_{l,i}(t)$ (for $i = 1, \ldots, r$) to obtain a model of the linear block using Subspace Methods.
Stop

Figure 6.4: Summary of the method for a system with p inputs and r outputs.

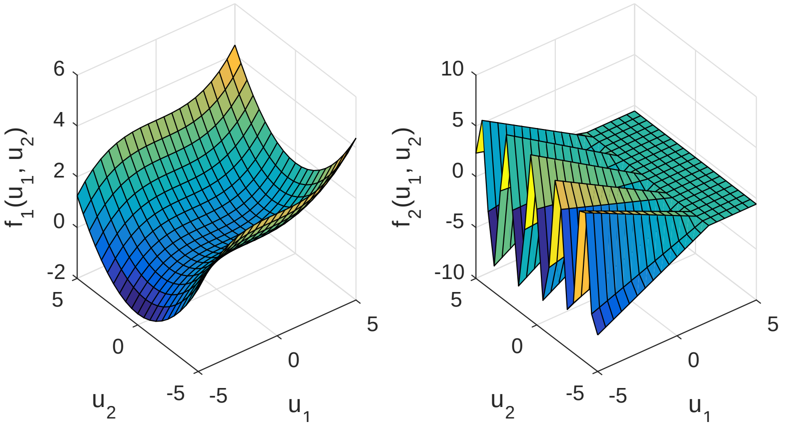


Figure 6.5: Nonlinear functions for Example 1. (Left) $f_1(u_1(t), u_2(t))$ and (Right) $f_2(u_1(t), u_2(t))$.

6.3 Experimental results and comparisons

The proposed method is applied to two examples. Each example has two inputs and two outputs and consists of two nonlinear functions and four Linear Time Invariant (LTI) blocks as illustrated in Fig. 6.1. Note that in order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D. The corresponding nonlinear functions of Example 1 are given in (6.6) and plotted in Fig. 6.5:

$$f_1(u_1, u_2) = \frac{u_1^3}{80} + \frac{0.9u_2^2}{8} \tag{6.6a}$$

$$f_2(u_1, u_2) = \begin{cases} (|u_1 - 2| \sin(3(u_2 - 2))) + d, \text{ for } u_1 \leq 2 \\ \hfill d, \text{ for } u_1 > 2, \end{cases} \tag{6.6b}$$

with $d = -0.5588$.

The transfer functions of Example 1 are presented in (6.7) and the magnitude of their frequency response is shown in Fig. 6.6:

$$G_{11}(q) = \frac{1.813}{q - 0.8187} \tag{6.7a}$$

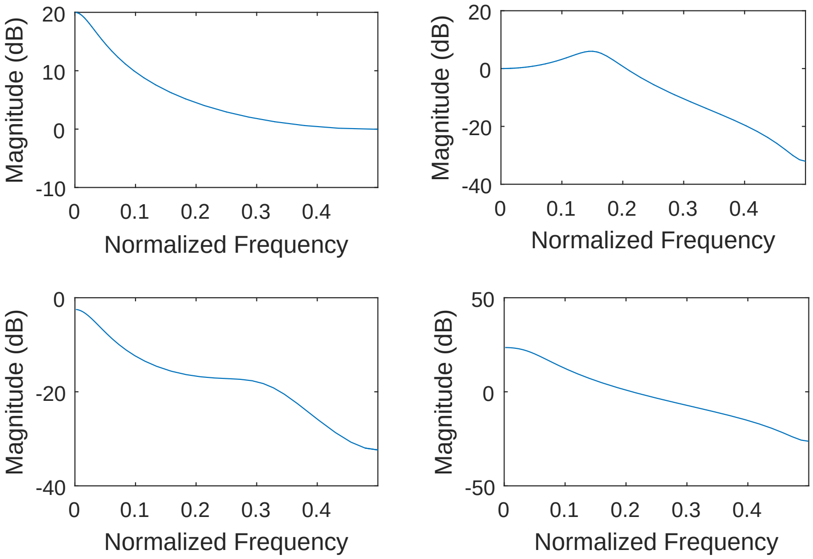


Figure 6.6: Magnitude of the frequency response of the LTI blocks for Example 1. (Up and left) $G_{11}(q)$, (Up and right) $G_{12}(q)$, (Down and left) $G_{21}(q)$ and (Down and right) $G_{22}(q)$.

$$G_{12}(q) = \frac{0.3929q + 0.3308}{q^2 - 0.8828q + 0.6065} \tag{6.7b}$$

$$G_{21}(q) = \frac{-0.045244(q + 1.668)(q - 1.646)(q + 0.2122)}{(q - 0.7408)^2(q^2 + 0.5048q + 0.3679)} \tag{6.7c}$$

$$G_{22}(q) = \frac{0.79928(q + 0.8185)}{(q^2 - 1.452q + 0.5488)}. \tag{6.7d}$$

For Example 2 the corresponding nonlinear functions are presented in (6.8) and plotted in Fig. 6.7:

$$f_1(u_1, u_2) = \frac{u_1^3}{5} + \sin(u_2)u_2^2 \tag{6.8a}$$

$$f_2(u_1, u_2) = 10\sin(u_1) + u_2^2. \tag{6.8b}$$

The transfer functions of Example 2 are given in (6.9) and the magnitude of their

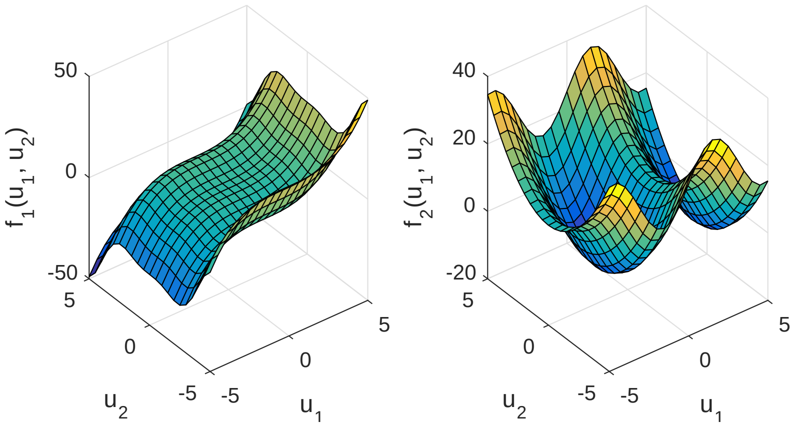


Figure 6.7: Nonlinear functions for Example 2. (Left) $f_1(u_1(t), u_2(t))$ and (Right) $f_2(u_1(t), u_2(t))$.

frequency response is shown in Fig. 6.8:

$$G_{11}(q) = \frac{100q^3 + 300q^2 + 300q + 100}{q^3 - 2.458q^2 + 2.262q - 0.7654} \tag{6.9a}$$

$$G_{12}(q) = \frac{18000q^2 - 32400q + 14400}{q^2 - 1.5q + 0.7225} \tag{6.9b}$$

$$G_{21}(q) = 1000\frac{q^4 - 1.884q^3 + 2.506q^2 - 1.884q + 1}{q^4 - 2.485q^3 + 2.528q^2 - 1.184q + 0.2245} \tag{6.9c}$$

$$G_{22}(q) = \frac{100q^3 - 50.64q^2 - 50.64q + 100}{q^3 - 2.564q^2 + 2.218q - 0.6456}. \tag{6.9d}$$

The results of 100 Monte Carlo simulations of the proposed method for different Signal to Noise Ratios (SNR) are presented in Figs. 6.9 and 6.10 for Examples 1 and 2 respectively.

The proposed method, from now on referred to as MIMO-H-STST, is compared with 3 other state of the art methods, namely:

- NARX LS-SVM (Suykens et al., 2002).

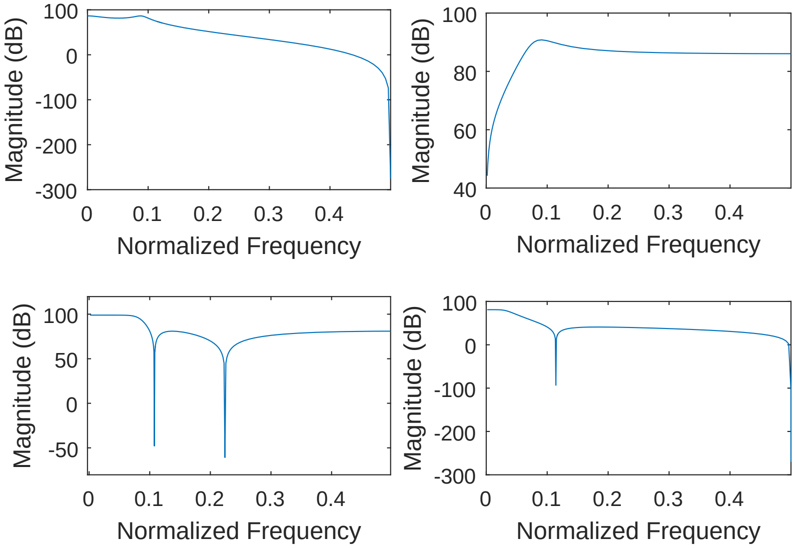


Figure 6.8: Magnitude of the frequency response of the LTI blocks for Example 2. (Up and left) $G_{11}(q)$, (Up and right) $G_{12}(q)$, (Down and left) $G_{21}(q)$ and (Down and right) $G_{22}(q)$.

- The method in Jeng and Huang (2008) where an approximation to the impulse response of the system is obtained and with it and the known outputs an estimation of the intermediate variables is found. Using this approximation and the known inputs, a mapping of the nonlinear block is done through the fitting of multivariate polynomials. From now on, this method will be referred to as IR H-MIMO.

- Using orthonormal bases for the identification of block oriented nonlinear systems is proposed in Gomez and Baeyens (2004). This method will be referred to as ONBF.

The results of 100 Monte Carlo simulations are summarized in Table 6.1 where for each of the methods mentioned above, the median is presented for different Signal to Noise Ratios.

In the proposed method for estimating the nonlinear part 900 points were used. To obtain those points, the length of the steps with the shortest duration for Example 1

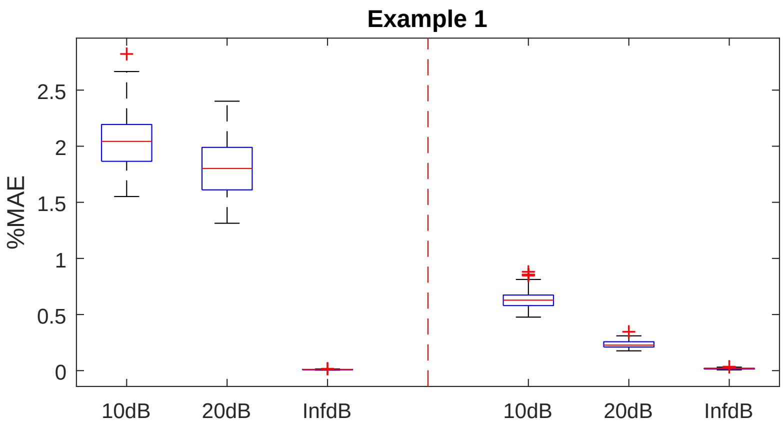


Figure 6.9: Results of 100 Monte Carlo simulations of the proposed method in Example 1. (Left) Output 1. (Right) Output 2.

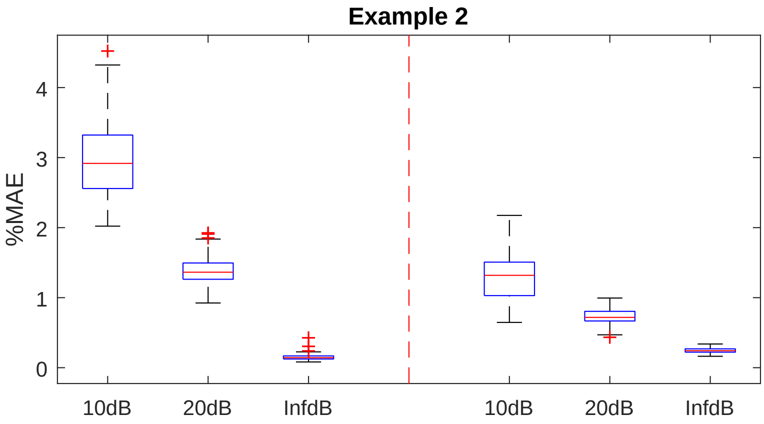


Figure 6.10: Results of 100 Monte Carlo simulations of the proposed method in Example 2. (Left) Output 1. (Right) Output 2.

Table 6.1: $\%MAE$ Comparison for the different methods tested. Medians are offered for 100 Monte Carlo simulations for each case.

		Example 1		**Example 2**	
		y_1	y_2	y_1	y_2
SNR 10dB	**MIMO-H-STST**	**2.0431**	**0.62828**	**2.9171**	**1.3187**
	NARX LS-SVM	9.2493	3.4636	14.0744	5.9171
	IR H-MIMO	13.1221	20.7751	15.5643	20.2475
	ONBF	12.2418	12.4902	2.954	6.8069
SNR 20dB	**MIMO-H-STST**	**1.8017**	**0.22772**	**1.3635**	**0.71816**
	NARX LS-SVM	5.6842	1.7901	13.6958	3.3421
	IR H-MIMO	10.4575	16.9498	3.1664	7.2392
	ONBF	10.9409	12.363	1.5856	4.3417
SNR InfdB	**MIMO-H-STST**	**0.008942**	**0.017892**	**0.1428**	**0.24264**
	NARX LS-SVM	4.1052	0.9849	13.6734	2.5123
	IR H-MIMO	9.6007	13.8155	0.23985	0.98531
	ONBF	10.7441	12.3495	1.3339	3.8984

was set to 50 samples, meaning that the whole time series used consisted of 45000 samples. For Example 2 the length of the steps with the shortest duration was fixed to 90 samples, thus the time series consisted of 81000 samples. In both examples Δ_T was set to 10 samples. The linear part was identified from a dataset with 4500 samples generated by applying Pseudo Random Multilevel Signals to the system and using the subspace method N4SID (Van Overschee & De Moor, 1996). The model order was selected by looking at the plot of the singular values of the Hankel matrices of the impulse response for different orders (from 1 to 10).

For the NARX LS-SVM approach a training set was generated using the combination of the amplitudes in the input signals used for the proposed method (i.e. $\tilde{\boldsymbol{U}}$). This means that 900 points were used for training the model. For the parameter tuning, Coupled Simulating Annealing Xavier-de Souza et al. (2009) followed by a Simplex approach was used under a 10 fold cross validation scheme. 10 lags of input and 10 lags of output were employed.

Pseudo Random Binary Signals (PRBS) of 800 samples were created in order to identify the linear part when using the IR H-MIMO method. In a first stage $u_1(t)$ was a PRBS and $u_2(t)$ was kept at 0. Then, in a second stage $u_1(t)$ was 0 and $u_2(t)$ was a PRBS. After the impulse responses were estimated, the nonlinear part was modeled. To do this, signals of 980 points were used of which the last 80 where included to make the corresponding linear system overdetermined. The initial 900 points where generated guaranteeing that all combinations of 30 points drawn from a uniform distribution between -5 and 5 were included. With these signals the nonlinearities were estimated by fitting two-dimensional polynomials with degrees 3 and 7 for Examples 1 and 2 respectively.

Polynomial basis functions were used for identifying the nonlinearity for the ONBF method. For Example 1 until degree 3 and for Example 2 until degree 5. It was found empirically that the use of simpler basis functions yielded better results for the modelling of the linear part, consequently q^{-n} was used. The number of bases used for Example 1 was 10 while for Example 2 was 40. These values were set by trial and error and were the ones that offered a good trade-off between complexity and accuracy. The number of data points used was 1600 for the first example and 3600 for the second one.

For the examples presented, the proposed method clearly outperforms the other methods considered. It is important to highlight that the nonlinearities in the examples used are very difficult to model using polynomial basis functions as they do not belong to the problem class. For the proposed method, which does not require previous knowledge about the problem class, this is not a problem at all.

It can be seen that the proposed method is robust against the type of noise employed, as the results remain good even when adding high levels of noise.

6.4 Conclusions

A new methodology for identifying MIMO Hammerstein systems is presented in this chapter. This method exploits the steady-state behavior of the system in order to approximate the nonlinear part. To do this the method profits from the good generalization capabilities of LS-SVM which allow it to deal with hard nonlinearities.

The proposed method is very flexible with respect to the number of inputs and outputs it can handle. However, for non-square systems the estimated model will have a different internal structure than the actual one.

The used examples show that the method has very good generalization capabilities and can work with different problem classes including systems with hard nonlinearities. This constitutes a nice advantage when the class of problem is unknown or is difficult to model with certain basis functions.

It is shown that the proposed method is robust against the type of noise employed as even in the presence of high levels of noise it has a good performance. In fact, for the examples presented it performed better than the other state of the art methods compared in this chapter.

Chapter 7

A two-experiment approach to Wiener system identification

7.1 Introduction

In this chapter, we discuss a novel method for Wiener systems that separates the estimation of the nonlinearity from the identification of the LTI block, facilitating the identification process and reducing the computational burden of maximum likelihood/prediction error techniques. To do so, it is required that the user has the freedom to design two separate experiments, each consisting of feeding the system with a specific input.

In the first experiment, the system is driven by a simple sinusoidal signal with prefixed frequency and phase. Using this signal, we show that we can easily reconstruct the static nonlinearity as a function of the unknown phase delay introduced by the LTI block. We discuss three possible modeling approaches for the nonlinearity. Depending on the adopted approach, we show how to fully recover the nonlinear function (up to a scaling factor), that is, how to remove the ambiguity introduced by the unknown phase delay.

This chapter is based on:
Bottegal, G., **Castro-Garcia, R.**, Suykens, J. A. K. (2017a). *On the identification of Wiener systems with polynomial nonlinearity*. Accepted for publication in the proceedings of the 56th IEEE Conference on Decision and Control (CDC 2017). (Internal Report 17-55, ESAT-SISTA, KU Leuven. Leuven, Belgium).
Bottegal, G., **Castro-Garcia, R.**, Suykens, J. A. K. (2017b). *A two-experiment approach to Wiener system identification*. In Internal report 17-38, ESAT-SISTA, KU Leuven (Leuven, Belgium).

The first modeling approach relies on a parametric description of the nonlinearity as a linear combination of a number of basis functions. Here, the phase delay is recovered using a special instance of separable least-squares. The second approach is a special case of the first, where the basis functions are monomials. In this case, the function can be fully estimated via a simple procedure involving least-squares estimation. The third approach is a nonparametric one; it relies on the least squares support vector machines (LS-SVM) framework (Suykens et al., 2002), under the assumption that the nonlinearity is a smooth function. In this case, the phase delay is estimated along with the hyperparameters characterizing the kernel used in the LS-SVM estimation procedure.

Using the estimated model of the static nonlinearity, we perform a second experiment where the system is fed with a persistently exciting input. In this way, we can identify the LTI block by means of a modified version of the standard prediction error method (PEM) for linear output-error (OE) systems (Ljung, 1999). The computational burden of this second step reduces essentially to the one of PEM for OE systems.

The proposed framework is tested via numerical experiments showing its effectiveness compared to other identification techniques for Wiener systems.

The chapter is organized as follows: In Section 7.2 we define the problem under study. Section 7.3 describes the proposed method including its parametric and nonparametric versions. Section 7.4 illustrates the results found when applying the described methodology on two simulation examples and compares the obtained results with other methods in the literature. Finally, in Section 7.5 some conclusions are presented.

7.2 Wiener system identification using a two experiment approach

We consider the following SISO system, also called a Wiener system (see Fig. 1.2 for a schematic representation):

$$\begin{aligned} x(t) &= G(q^{-1})u(t) \\ y(t) &= f(x(t)) + e(t)\,. \end{aligned} \tag{7.1}$$

In the former equation, $G(q^{-1})$ represents the transfer function of a causal LTI subsystem, driven by the input $u(t)$. In the latter equation, $y(t)$ is the result of a static nonlinear transformation, denoted by $f(\cdot)$, of the signal $x(t)$, and $e(t)$ is white noise with unknown variance σ^2. The problem under study is to estimate the LTI subsystem and the nonlinear function from a set of input and output measurements.

We assume that the user has the freedom to design the input signal $u(t)$. In particular, we assume that the user has the possibility to run two separate experiments, each having a particular signal $u(t)$ as an input. The goal of this chapter is to describe an identification technique for the system (7.1) that is linked to a particular choice of these experiments. It consists of the two following steps:

1. Feed the system with a sinusoid at a prescribed frequency. Use the steady-state data to estimate the nonlinear function $f(\cdot)$.
2. Feed a system with a persistently exciting input signal and identify the LTI subsystem using the information gathered on the first step regarding the static nonlinearity.

Let us first briefly discuss the second step of the proposed procedure. Let

$$G(q^{-1}) = \frac{b_0 + b_1 q^{-1} + \ldots + b_m q^{-m}}{1 + a_1 q^{-1} + \ldots + a_n q^{-n}}\,, \tag{7.2}$$

so that the LTI subsystem is completely characterized by the parameter vector $\boldsymbol{\theta} := \begin{bmatrix} b_0 & b_1 & \ldots & b_m & a_1 & \ldots & a_n \end{bmatrix}$. Then, assuming that an estimate of the nonlinearity say, $\hat{f}(\cdot)$, is available after the first step of the procedure, we can set up a PEM-based identification criterion as follows

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N_2} \sum_{t=1}^{N_2} \Big(y(t) - \hat{f}(G(q^{-1})u(t)) \Big)^2 \,, \tag{7.3}$$

where N_2 is the number of samples collected during the second experiment. Note that this is a mild generalization of the standard PEM, requiring only to account, in the optimization process, for the nonlinear transformation induced by $\hat{f}(\cdot)$. This does not make the solution of (7.3) harder than a standard PEM applied to an output-error model, because in both cases we have to face a nonlinear optimization problem, and in both cases gradient-based methods can be easily applied (Ljung, 1999). Moreover, it has been shown in Wigren (1994) that recursive PEM schemes for Wiener system identification with known nonlinearity are guaranteed to converge under mild assumptions.

As opposed to the aforementioned second step, the first step can be more involved and requires a more thorough analysis. We shall focus on this step in the remainder of the chapter.

Remark 1. The method can be easily extended to the case where the additive noise is colored. In that case, the parameters of the noise shaping filter can be estimated either in the first or in the second phase of the proposed procedure.

7.3 Three approaches to estimate the nonlinearity

In this section we discuss the first step of the procedure, proposing three estimation approaches for the static nonlinearity.

We consider the following input signal

$$u(t) = \sin(\omega t + \phi_0)\,,$$

where ω is an user-prescribed frequency and ϕ_0 is a known phase delay. Then, after the transient effect of $G(q^{-1})$ has vanished, we have that

$$x(t) = A_\omega \sin(\omega t + \phi_0 + \phi_\omega)\,,$$

where A_ω and ϕ_ω are the gain and the phase delay of the LTI subsystem $G(q^{-1})$ at the frequency ω (Ljung, 1999, Ch. 2). Due to the structural non-identifiability of Wiener systems, A_ω can not be determined (see Remark 2 below). We thus drop it and define a new signal

$$\bar{x}(t) = \sin(\omega t + \phi_0 + \phi_\omega)\,,$$

which is parameterized by the unknown quantity ϕ_ω. Accordingly, we write the output of the system as

$$y(t) = f(\sin(\omega t + \phi_0 + \phi_\omega)) + e(t)\,. \tag{7.4}$$

Then, the problem under study, that is to estimate $f(\cdot)$, is coupled with the problem of estimating ϕ_ω. In the following, we describe three approaches to this problem, assuming that the number of collected samples of $y(t)$ (at its steady state) is equal to N_1.

Remark 2. Since we are estimating the static nonlinearity using the signal $\bar{x}(t)$ instead of $x(t)$, we are obtaining a scaled (in the x-axis) version of $f(\cdot)$, that is, we are estimating $f(x/A_\omega)$ instead of $f(x)$. This scaling effect is compensated in the second stage of the method; in fact (7.3) will return the estimate $A_\omega G(q^{-1})$ instead of $G(q^{-1})$. Then, we need additional information (e.g., on the LTI system gain, see Bai (1998)) to uniquely recover $G(q^{-1})$ and $f(\cdot)$; this lack of identifiability is a well known issue in block oriented system identification. However, if the focus is on output prediction (as is in the experiments of Section 7.4), rescaling of the two blocks is not required.

7.3.1 Parametric approach

We assume that there exist a set of known basis functions $h_0(\cdot),\ h_1(\cdot),\ \ldots,\ h_p(\cdot)$ such that

$$f(x) = \sum_{i=0}^{p} c_i h_i(x) \quad , \quad \forall x \in \mathbb{R}\,.$$

Then, the problem of estimating $f(\cdot)$ reduces to determining the coefficients $c_0, \ldots, c_p$. We rewrite (7.4) as

$$y(t) = \sum_{i=0}^{p} c_i h_i(\sin(\omega t + \phi_0 + \phi_\omega)) + e(t)\,. \tag{7.5}$$

Let $\boldsymbol{H}(\phi_\omega) \in \mathbb{R}^{N_1 \times p+1}$ be a matrix such that

$$\boldsymbol{H}(\phi_\omega)_{t,i} = h_i(\sin(\omega t + \phi_0 + \phi_\omega))\,.$$

Then we have the following regression model

$$y = \boldsymbol{H}(\phi_\omega)\boldsymbol{c} + e\,, \tag{7.6}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{N_1} \end{bmatrix} \quad , \quad \boldsymbol{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_{N_1} \end{bmatrix} \quad , \quad \boldsymbol{c} = \begin{bmatrix} c_0 \\ \vdots \\ c_p \end{bmatrix} \,. \tag{7.7}$$

Because ϕ_ω is unknown, we treat it as an unknown parameter to be determined together with $\boldsymbol{c}$; for sake of clarity we rewrite the regression model (7.6) replacing ϕ_ω with a generic ϕ:

$$\boldsymbol{y} = \boldsymbol{H}(\phi)\boldsymbol{c} + \boldsymbol{e}\,. \tag{7.8}$$

An unbiased estimate of $\boldsymbol{c}$ can be obtained via least-squares (recall that $\boldsymbol{e}$ is white noise). This estimate is function of the unknown phase delay introduced by the LTI block, and corresponds to

$$\widehat{\boldsymbol{c}(\phi)} = \left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1} \boldsymbol{H}^\top(\phi)\boldsymbol{y}\,. \tag{7.9}$$

To estimate ϕ (and hence $\boldsymbol{c}$), we introduce the following criterion:

$$\hat{\phi}_\omega = \arg\min_{\phi \in \mathcal{I}} \frac{1}{N_1} \|\boldsymbol{y} - \hat{\boldsymbol{f}}_\phi\|^2\,, \tag{7.10}$$

where $\hat{\boldsymbol{f}}_\phi = \boldsymbol{H}(\phi)\hat{\boldsymbol{c}}(\phi)$ is the estimate of $f(\cdot)$, evaluated at $\bar{x}(t)$, $t = 1, \ldots, N_1$ and in vector notation, and $\mathcal{I}$ is a suitable subset of the negative real semi-axis (recall that the phase delay is always negative for causal systems). The question is under which conditions (7.10) admits an unique solution. To this end we introduce the following concept.

Definition 1. *A function $h(\cdot)$ is* phase-indistinguishable *in the set $\mathcal{I}$ if there exist ϕ_1 and ϕ_2 in $\mathcal{I}$ and $a \in \mathbb{R}$ such that*

$$h(\sin(\omega t + \phi_1)) = a h(\sin(\omega t + \phi_2))\,, \tag{7.11}$$

for all $t \in \mathbb{N}$.

Therefore, if $f(\cdot)$ is parameterized through a set of phase-indistinguishable basis functions, criterion (7.10) may not admit a unique minimum. Quite fortunately, the following lemma clarifies that, with a suitable definition of $\mathcal{I}$, the set of phase-indistinguishable functions becomes trivial (the proof is direct and thus it is skipped).

Lemma 1. *Condition* (7.11) *is satisfied if and only if $\phi_2 = \phi_1 + k\pi$, $k \in \mathbb{Z}$, (and $a = (-1)^k$) or h is constant (and $a = 1$).*

Therefore, we have to restrict our phase search in the interval $\mathcal{I} = (-\pi,\, 0]$. Using the lemma, we can prove the following result:

Proposition 1. *Let $\mathcal{I} = (-\pi,\, 0]$. It holds that*

$$\phi_\omega = \arg\min_{\phi\in\mathcal{I}} \mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{f}}_\phi\|^2 \,.$$

Similarly, we show that the phase estimation is consistent.

Proposition 2. *Let $e(t)$ be a stochastic stationary ergodic process with finite variance. Then the asymptotic (in N_1) solution of the problem* (7.10) *is equal to ϕ_ω.*

Proof. We have

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{y}-\hat{\boldsymbol{f}}_\phi\|^2 &= \mathbb{E}\|y - \boldsymbol{H}(\phi)\hat{\boldsymbol{c}}(\phi)\|^2 \\
&= \mathbb{E}\left\|\boldsymbol{y} - \boldsymbol{H}(\phi)\left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1}\boldsymbol{H}^\top(\phi)\boldsymbol{y}\right\|^2 \\
&= \mathbb{E}\left\|\left(\boldsymbol{I} - \boldsymbol{H}(\phi)\left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1}\boldsymbol{H}^\top(\phi)\right)\boldsymbol{y}\right\|^2 \\
&= \mathbb{E}\left\|\left(\boldsymbol{I} - \boldsymbol{H}(\phi)\left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1}\boldsymbol{H}^\top(\phi)\right)\right. \\
&\qquad \left.\times\left(\boldsymbol{H}(\phi_\omega)\boldsymbol{c} + \boldsymbol{e}\right)\right\|^2 \\
&= \left\|\left(\boldsymbol{I}-\boldsymbol{H}(\phi)\left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1}\boldsymbol{H}^\top(\phi)\right)\boldsymbol{H}(\phi_\omega)\boldsymbol{c}\right\|^2 \\
&\quad + \sigma^2\mathrm{tr}\left[\left(\boldsymbol{I}-\boldsymbol{H}(\phi)\left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1}\boldsymbol{H}^\top(\phi)\right)^2\right].
\end{aligned}$$

Now, it is easy to see that, since

$$\left(\boldsymbol{I} - \boldsymbol{H}(\phi)\left(\boldsymbol{H}^\top(\phi)\boldsymbol{H}(\phi)\right)^{-1}\boldsymbol{H}^\top(\phi)\right)$$

is a projection matrix, its trace is constant (and equal to its rank), independently of the value assumed by ϕ. Hence, it does not affect the value of $\mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{f}}_\phi\|^2$. As for the first term of the above equality, we have

$$\left\| \left(\boldsymbol{I} - \boldsymbol{H}(\phi) \left(\boldsymbol{H}^\top(\phi) \boldsymbol{H}(\phi) \right)^{-1} \boldsymbol{H}^\top(\phi) \right) \boldsymbol{H}(\phi_\omega) \boldsymbol{c} \right\|^2 \geq 0 \,. \qquad (7.12)$$

Equality to 0 holds only when $\boldsymbol{H}(\phi) = \boldsymbol{H}(\phi_\omega)$, because in $[0, \pi)$ there always exist at least one of the basis functions that is not phase-indistinguishable. This concludes the proof of Proposition 1. By multiplying the objective function of (7.10) by $1/N_1$, and letting $N_1 \to \infty$ we have

$$\frac{1}{N_1} \sum_{t=1}^{N_1} (y(t) - \hat{f}_\phi(x(t)))^2 \to \mathbb{E}\|y(t) - \hat{f}_\phi(x(t))\|^2 \,, \qquad (7.13)$$

where the expectation is taken with respect to the distribution of $e(t)$. The result of Proposition 2 follows as a special case of Proposition 1. □

Algorithm 2 summarizes the first step of the proposed procedure, under a parametric modeling assumption of the static nonlinearity. Albeit (7.10) is still a nonlinear problem, it requires to search for the optimum of a scalar variable within a bounded interval. Therefore, it can be solved quickly by using e.g. a grid search over $\mathcal{I}$.

Algorithm 2 Parametric static nonlinearity estimation

```
Input: $\{y(t)\}_{t=1}^{N_1}$, $\omega$, $\phi_0$
Output: $\hat{c}_0, \ldots, \hat{c}_p$
1: Select a subset of values $\phi \in \mathcal{I}$.
2: For every $\phi$ solve (7.9) and evaluate $\|\boldsymbol{y} - \hat{\boldsymbol{f}}_\phi\|^2$
3: Choose $\hat{\boldsymbol{c}}(\phi)$ associated with $\phi$ solving (7.10)
```

7.3.2 Polynomial nonlinearity

We now discuss the case where the nonlinear function $f(\cdot)$ can be expressed as a polynomial of known order, namely

$$f(x) = c_0 + c_1 x + \ldots + c_p x^p \,. \qquad (7.14)$$

For ease of exposition, we assume that p is even (the case p odd runs along the same lines of reasoning). We recall that (see e.g. Beyer (1978), Novak, Simon, Kadlec, and

Lotton (2010)), for $i \in \mathbb{N}$,

$$\sin^{2i}(x) = \frac{1}{2^{2i}} \begin{pmatrix} 2i \\ i \end{pmatrix} + \frac{(-1)^i}{2^{2i-1}} \sum_{k=0}^{i-1} (-1)^k \begin{pmatrix} 2i \\ k \end{pmatrix} \cos(2(i-k)x)\,,$$

$$\sin^{2i+1}(x) = \frac{(-1)^i}{4^i} \sum_{k=0}^{i} (-1)^k \begin{pmatrix} 2i+1 \\ k \end{pmatrix} \sin((2i+1-2k)x)\,,$$

and that, for any $r \in \mathbb{Z}$,

$$\sin(r(wt+\phi_0+\phi_\omega)) = \cos(r\phi_\omega)\sin(r(wt+\phi_0)) + \sin(r\phi_\omega)\cos(r(wt+\phi_0))$$

$$\cos(r(wt+\phi_0+\phi_\omega)) = -\sin(r\phi_\omega)\sin(r(wt+\phi_0)) + \cos(r\phi_\omega)\cos(r(wt+\phi_0))\,.$$

Then, we can rewrite the nonlinear function as

$$\begin{aligned} f(\bar{x}(t)) = \sum_{i=0}^{p/2} c_{2i} \Big[\frac{1}{2^{2i}} \begin{pmatrix} 2i \\ i \end{pmatrix} &+ \frac{(-1)^i}{2^{2i-1}} \sum_{k=0}^{i-1} (-1)^k \begin{pmatrix} 2i \\ k \end{pmatrix} \\ &\times [\cos(2(i-k)\phi_\omega)\cos(2(i-k)(wt+\phi_0)) \\ &-\sin(2(i-k)\phi_\omega)\sin(2(i-k)(wt+\phi_0))] \Big] \\ &+ \sum_{i=0}^{p/2-1} c_{2i+1} \frac{(-1)^i}{4^i} \sum_{k=0}^{i} (-1)^k \begin{pmatrix} 2i+1 \\ k \end{pmatrix} \\ &\times [\cos((2i{+}1{-}2k)\phi_\omega)\sin((2i{+}1{-}2k)(wt{+}\phi_0)) \\ &+\sin((2i{+}1\,{-}2k)\phi_\omega)\cos((2i{+}1\,{-}2k)(wt{+}\phi_0))]\,. \end{aligned} \tag{7.15}$$

This equation is particularly interesting because it permits to express $f(\bar{x}(t))$ as a linear combination of sines and cosines with frequency $r\omega(t)$ and known phase delay $r\phi_0$, where $r = 0, \ldots, p$. The coefficients of this linear combination are also a function of sines and cosines of the unknown phase delay ϕ_ω. Let

$$k_0 := \sum_{i=0}^{p/2} c_{2i} \frac{1}{2^{2i}} \binom{2i}{i}$$

$$k_{s,r} := \sum_{i=r/2}^{p/2} c_{2i} \frac{1}{2^{2i-1}} \binom{2i}{i-r/2} \sin(r\phi_\omega) \qquad (r \text{ even})$$

$$k_{c,r} := -\sum_{i=r/2}^{p/2} c_{2i} \frac{1}{2^{2i-1}} \binom{2i}{i-r/2} \cos(r\phi_\omega) \qquad (r \text{ even})$$

$$k_{s,r} := (-1)^{2i-(r-1)/2} \sum_{i=(r-1)/2}^{p/2-1} c_{2i+1} \frac{1}{4^i} \binom{2i+1}{i-(r-1)/2} \cos(r\phi_\omega) \ (r \text{ odd})$$

$$k_{c,r} := (-1)^{2i-(r-1)/2} \sum_{i=(r-1)/2}^{p/2-1} c_{2i+1} \frac{1}{4^i} \binom{2i+1}{i-(r-1)/2} \sin(r\phi_\omega) \ (r \text{ odd})$$

Let also, for $r = 1, \dots, p$,

$$k_r = \sum_{i=r/2}^{p/2} c_{2i} \frac{1}{2^{2i-1}} \binom{2i}{i-r/2}, \qquad (r \text{ even})$$

$$k_r = \sum_{i=(r-1)/2}^{p/2-1} c_{2i+1} \frac{1}{4^i} \binom{2i+1}{i-(r-1)/2}, \qquad (r \text{ odd})$$

so that there exists a matrix $\boldsymbol{M} \in \mathbb{R}^{p+1 \times p+1}$ such that

$$\mathbf{k} = \boldsymbol{M}\boldsymbol{c}, \quad \mathbf{k} := \begin{bmatrix} k_0 & \dots & k_p \end{bmatrix}^\top . \tag{7.16}$$

Then we can write

$$\bar{\mathbf{k}} = \begin{bmatrix} k_0 \\ k_{s,1} \\ k_{c,1} \\ \vdots \\ k_{s,p} \\ k_{c,p} \end{bmatrix} = \begin{bmatrix} \gamma_0 k_0 \\ \gamma_{s,\,1} k_1 \sin(\phi_\omega) \\ \gamma_{c,\,1} k_1 \cos(\phi_\omega) \\ \vdots \\ \gamma_{s,\,p} k_p \sin(p\phi_\omega) \\ \gamma_{c,\,p} k_p \cos(p\phi_\omega) \end{bmatrix}, \tag{7.17}$$

where γ_0 and the $\gamma_{s,i}$, $\gamma_{c,i}$, $i = 1, \ldots, p$, are equal to -1 or 1 and are known in advance. Therefore, if we are able to estimate the vector $\bar{\mathbf{k}}$ and the phase delay ϕ_ω, we can also estimate $\mathbf{k}$ and consequently $\boldsymbol{c}$. To this end, we define

$$\begin{aligned}\boldsymbol{\psi}(t)^\top := [1 \;\; \cos(\omega t + \phi_0) \;\; \sin(\omega t + \phi_0) \;\; \ldots \\ \ldots \;\; \sin(p(\omega t + \phi_0)) \;\; \cos(p(\omega t + \phi_0))] \,,\end{aligned} \tag{7.18}$$

so that we can rewrite (7.15) as $f(\bar{x}(t)) = \boldsymbol{\psi}(t)^\top \bar{\mathbf{k}}$, and express the measurement equation via the linear regression model

$$y(t) = \boldsymbol{\psi}(t)^\top \bar{\mathbf{k}} + e(t) \,.$$

We can then obtain the least-squares estimate

$$\widehat{\bar{\mathbf{k}}} = \left(\sum_{t=1}^{N_1} \boldsymbol{\psi}(t)\boldsymbol{\psi}(t)^\top \right)^{-1} \sum_{t=1}^{N_1} \boldsymbol{\psi}(t) y(t) \,. \tag{7.19}$$

Using this estimate, we now show how to recover the coefficients of the polynomial. From (7.17), the absolute value of each k_i, $i = 1, \ldots, p$, can be reconstructed via

$$|\hat{k}_i| = \sqrt{\hat{k}_{s,i}^2 + \hat{k}_{c,i}^2} \,. \tag{7.20}$$

Using the estimates $\hat{k}_{s,1}$ and $\hat{k}_{c,1}$, one can recover the phase delay ϕ_ω as

$$\hat{\phi}_\omega = \tan^{-1}\left(\frac{\hat{k}_{s,1}}{\hat{k}_{c,1}} \right) \,, \tag{7.21}$$

where uniqueness of the solution is guaranteed in $\mathcal{I} = (-\pi, 0]$. Using $\hat{\phi}_\omega$, we can uniquely recover the sign of the coefficients k_i, exploiting the knowledge of the coefficients $\gamma_{s,i}$, $\gamma_{c,i}$ introduced in (7.17). Finally, the estimate $\hat{\boldsymbol{c}}$ can be recovered via

$$\hat{\boldsymbol{c}} = \boldsymbol{M}^{-1} \hat{\mathbf{k}} \,, \tag{7.22}$$

where $\boldsymbol{M}$ is defined in (7.16).

We summarize the procedure for the polynomial case in Algorithm 3.

We observe that this procedure still requires to recover the phase delay ϕ_ω. However, in this case this can be done with one simple operation, namely (7.21), instead of requiring the solution of a dedicated optimization problem, like in the general parametric case discussed in Section 7.3.1. Furthermore, it is not required to have a highly accurate estimate of ϕ_ω. In fact, what we need is just that $\hat{\phi}_\omega$ lies in the same orthant of ϕ_ω, so that we can correctly determine the sign of the coefficients k_i, $i = 0, \ldots, p$. As for the asymptotic performance of this estimation procedure, we have the following result showing that the asymptotic variance of the procedure *does not depend on the choice of* ω.

Algorithm 3 Polynomial static nonlinearity estimation

```
Input: ${y(t)}_{t=1}^{N_1}$, $\omega$, $\phi_0$
Output: $\hat{c}_0, \ldots, \hat{c}_p$
1: Construct the regressors (7.18) and compute the least-squares estimate (7.19)
2: Recover the absolute value of the coefficients $k_i$, $i = 0, \ldots, p$ using (7.20)
3: Estimate the phase shift via (7.21) and the sign of the coefficients $k_i$, $i = 0, \ldots, p$
4: Recover the coefficients $c_i$, $i = 0, \ldots, p$, using (7.22)
```

Proposition 3. *The procedure outlined above gives consistent estimates of the coefficients c_i, $i = 0, \ldots, p$. Furthermore, the asymptotic covariance of their estimates is equal to*

$$\frac{\sigma^2}{N_1}\boldsymbol{M}^{-1}\boldsymbol{D}\boldsymbol{M}^{-T}\,, \tag{7.23}$$

where $\boldsymbol{D} = \operatorname{diag}\{1,\, 2,\, \ldots,\, 2\}$.

Proof. Let

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_0 & 0 & & \cdots & 0 \\ 0 & \gamma_{s,\,1}\sin(\phi_\omega) & 0 & \cdots & 0 \\ 0 & \gamma_{c,\,1}\cos(\phi_\omega) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & & 0 & \gamma_{s,\,p}\sin(p\phi_\omega) \\ 0 & \cdots & & 0 & \gamma_{c,\,p}\cos(p\phi_\omega) \end{bmatrix}, \tag{7.24}$$

so that (7.17) can be rewritten as $\bar{\mathbf{k}} = \mathbf{\Gamma}\mathbf{k}$. We note that the Moore-Penrose pseudoinverse of $\mathbf{\Gamma}$ is

$$\mathbf{\Gamma}^{\#} = (\mathbf{\Gamma}^\top\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^\top = \mathbf{\Gamma}^\top\,,$$

so that $\mathbf{k} = \mathbf{\Gamma}^\top\bar{\mathbf{k}}$.

Since we do not need to estimate ϕ_ω but only its sign, we can conclude that the covariance matrix of the estimated coefficients c_i, $i = 0, \ldots, p$ can be calculated as the same as the one obtained via least-squares, that is

$$\begin{aligned} \mathbb{E}[(\hat{\boldsymbol{f}} - \boldsymbol{c})(\hat{\boldsymbol{f}} - \boldsymbol{c})^\top] &= \boldsymbol{M}^{-1}\mathbb{E}[(\hat{\mathbf{k}} - \mathbf{k})(\hat{\mathbf{k}} - \mathbf{k})^\top]\boldsymbol{M}^{-T} \\ &= \boldsymbol{M}^{-1}\mathbf{\Gamma}^\top\mathbb{E}[(\widehat{\bar{\mathbf{k}}} - \bar{\mathbf{k}})(\widehat{\bar{\mathbf{k}}} - \bar{\mathbf{k}})^\top]\mathbf{\Gamma}\boldsymbol{M}^{-T} \\ &= \sigma^2\boldsymbol{M}^{-1}\mathbf{\Gamma}^\top\left(\sum_{t=1}^{N_1}\boldsymbol{\psi}(t)\boldsymbol{\psi}(t)^\top\right)^{-1}\mathbf{\Gamma}\boldsymbol{M}^{-T}. \end{aligned} \tag{7.25}$$

Recalling the structure of $\boldsymbol{\psi}(t)$ given in (7.18), it is straightforward to check that, as N_1 grows large,

$$\frac{1}{N_1}\sum_{t=1}^{N_1}\boldsymbol{\psi}(t)\boldsymbol{\psi}(t)^\top \longrightarrow \text{diag}\left\{1,\, \frac{1}{2},\, \ldots,\, \frac{1}{2}\right\} := \bar{\boldsymbol{D}}^{-1}\,, \tag{7.26}$$

where $\bar{\boldsymbol{D}}$ has size $2p+1 \times 2p+1$. Equation (7.23) follows from the fact that

$$\boldsymbol{\Gamma}^\top \bar{\boldsymbol{D}} \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^\top \boldsymbol{\Gamma} \boldsymbol{D} = \boldsymbol{D}\,.$$

The consistency of the estimates follows from the consistency of the least-squares estimates (7.19). This concludes the proof. □

Example 1. Consider the third-order polynomial nonlinearity

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3\,.$$

We have

$$\begin{aligned} f(\bar{x}(t)) = c_0 + \frac{c_2}{2} &+ (c_1 + \frac{3}{4}c_3)\cos(\phi_\omega)\sin(\omega t + \phi_0) \\ &+ (c_1 + \frac{3}{4}c_3)\sin(\phi_\omega)\cos(\omega t + \phi_0) \\ &+ \frac{c_2}{2}\sin(2\phi_\omega)\sin(2(\omega t + \phi_0)) \\ &- \frac{c_2}{2}\cos(2\phi_\omega)\cos(2(\omega t + \phi_0)) \\ &- \frac{c_3}{4}\cos(3\phi_\omega)\sin(3(\omega t + \phi_0)) \\ &- \frac{c_3}{4}\sin(3\phi_\omega)\cos(3(\omega t + \phi_0))\,, \end{aligned}$$

and

$$\begin{bmatrix}\gamma_0\\ \gamma_{s,1}\\ \gamma_{c,1}\\ \gamma_{s,2}\\ \gamma_{c,2}\\ \gamma_{s,3}\\ \gamma_{c,3}\end{bmatrix} = \begin{bmatrix}1\\1\\1\\1\\-1\\-1\\-1\end{bmatrix}, \quad M = \begin{bmatrix}1 & 0 & 1/2 & 0\\ 0 & 1 & 0 & 3/4\\ 0 & 0 & 1/2 & 0\\ 0 & 0 & 0 & 1/4\end{bmatrix}.$$

The asymptotic covariance of the coefficient estimates, computed via (7.23), is then

$$\frac{\sigma^2}{N_1}\begin{bmatrix}3 & 0 & -4 & 0\\ 0 & 20 & 0 & -24\\ -4 & 0 & 8 & 0\\ 0 & -24 & 0 & 32\end{bmatrix},$$

which shows that, for the case $p = 3$, the odd coefficients are estimated with lower accuracy. Not surprinsingly, the estimates of the even coefficients and the estimates of the odd coefficients are uncorrelated.

7.3.3 Nonparametric approach

The obtained estimate depends on the regularization parameter γ entering (1.28), the kernel shaping hyperparameter in (1.29) (in this chapter the kernel parameter will be denoted as η to avoid confusion), and also the unknown phase delay ϕ_ω. Tuning suitable values for the first two quantities can be done using standard techniques, such as marginal likelihood, SURE, or cross validation (Friedman, Hastie, & Tibshirani, 2001). As for ϕ_ω, it is natural to ask to which interval we should restrict our search, and if we can treat it as an additional hyperparameter, to be tuned along with the other model hyperparameters. The following lemma answers the first question.

Lemma 2. *Let $K(\cdot, \cdot)$ be an isotropic kernel, i.e. a kernel such that $K(\bar{x}_i, \bar{x}_j) = K(|\bar{x}_i - \bar{x}_j|)$, for every $\bar{x}_i$, $\bar{x}_j$. Denote by $\hat{f}_{\phi_1}(\tilde{x})$ and $\hat{f}_{\phi_2}(\tilde{x})$ the estimates of $f(\cdot)$ at the point $\tilde{x}$ obtained using $\bar{x}(t) = \sin(\omega t + \phi_0 + \phi_1)$ and $\bar{x}(t) = \sin(\omega t + \phi_0 + \phi_2)$ as input locations, respectively. Then $\hat{f}_{\phi_1}(\tilde{x}) = \hat{f}_{\phi_2}(\tilde{x})$ for all $\tilde{x} \in \mathbb{R}$ if $\phi_1 = \phi_2 + k\pi$, $k \in \mathbb{Z}$.*

Proof Follows from the fact that

$$\sin(\omega t + \phi_0 + \phi_1) = \begin{cases} \sin(\omega t + \phi_0 + \phi_1 + k\pi) & \text{k even} \\ -\sin(\omega t + \phi_0 + \phi_1 + k\pi) & \text{k odd} \end{cases} \tag{7.27}$$

and the fact that the kernel is isotropic. □

Then, also in this case the search of ϕ_ω has to be restricted to $\mathcal{I} = (-\pi, 0]$. The following example clarifies that ϕ_ω can be seen as an additional kernel hyperparameter.

Example 2. Consider the RBF kernel (1.29). We have that

$$\begin{aligned}
(\bar{x}_i - \bar{x}_j)^2 &= (\sin(\omega i + \phi_0 + \phi_\omega) - \sin(\omega j + \phi_0 + \phi_\omega))^2 \\
&= (\sin\phi_\omega \cos(\omega i + \phi_0) + \cos\phi_\omega \sin(\omega i + \phi_0) \\
&\quad - \sin\phi_\omega \cos(\omega j + \phi_0) + \cos\phi_\omega \sin(\omega j + \phi_0))^2 \\
&= (\sin\phi_\omega (\cos(\omega i + \phi_0) - \cos(\omega j + \phi_0)) \\
&\quad + \cos\phi_\omega (\sin(\omega i + \phi_0) - \sin(\omega j + \phi_0)))^2 \\
&= [z_i^s - z_j^s \;\; z_i^c - z_j^c] \boldsymbol{\Psi} \begin{bmatrix} z_i^s - z_j^s \\ z_i^c - z_j^c \end{bmatrix}, \qquad (7.28)
\end{aligned}$$

where $z_i^s = \sin(\omega i + \phi_0)$, $z_i^c = \cos(\omega i + \phi_0)$ (the same notation holds for j) and

$$\boldsymbol{\Psi} = \begin{bmatrix} \cos^2\phi_\omega & \frac{1}{2}\sin 2\phi_\omega \\ \frac{1}{2}\sin 2\phi_\omega & \sin^2\phi_\omega \end{bmatrix}.$$

Then we can write

$$K(\bar{x}_i, \bar{x}_j) = \exp\left(\frac{-\|z_i - z_j\|_{\boldsymbol{\Psi}}^2}{\eta} \right),$$

with $z_i = [z_i^s \; z_i^c]^\top$, which shows that the RBF kernel with the input $\bar{x}(t) = \sin(\omega t + \phi_0 + \phi_\omega)$ can be seen as an RBF kernel with a bi-dimensional known input

$$z(t) = [\sin(\omega t + \phi_0) \;\; \cos(\omega t + \phi_0)],$$

weighted by the Mahalanobis distance $\boldsymbol{\Psi}$ (Weinberger & Tesauro, 2007), which depends on the hyperparameter ϕ_ω. Therefore, we can treat the tuning of ϕ_ω in the same way we treat the tuning of the other kernel hyperparameter η.

We summarize the procedure for nonparametric estimation of $f(\cdot)$ in Algorithm 4.

Algorithm 4 Nonparametric static nonlinearity estimation

```
Input: ${y(t)}_{t=1}^{N_1}$, $\omega$, $\phi_0$
Output: $f(\tilde{x})$, for any $\tilde{x} \in \mathbb{R}$
Tune the hyperparameters $\gamma$, $\eta$, $\phi_\omega$
Compute $f(\tilde{x})$ using (1.33)
```

7.4 Numerical experiments

The proposed methodology is tested on two simulated Wiener systems, which we refer to as S1 and S2. The LTI block of S1 is obtained using the Matlab command `cheby2(3,5,0.2)`, which returns a third-order system with stopband edge frequency 0.2π and 5 dB of stopband attenuation at the passband value. The static nonlinearity is the third-order polynomial $f(x) = x^3$. The parameters characterizing S1 are then

$$\begin{aligned} \boldsymbol{a} &= \begin{bmatrix} 2.46 & 2.26 & -0.77 \end{bmatrix} , \\ \boldsymbol{b} &= 10^{-2} \begin{bmatrix} 0.47 & 1.42 & 1.42 & 0.47 \end{bmatrix} , \\ \boldsymbol{c} &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} . \end{aligned}$$

As for S2, the LTI part is obtained using the Matlab command `cheby2(4,18,0.2)`, while the nonlinear part is the third-order polynomial $f(x) = x + 5x^2 - 0.5x^3$. Thus, S2 is described by the parameters

$$\begin{aligned} \boldsymbol{a} &= \begin{bmatrix} -2.49 & 2.53 & -1.18 & 0.22 \end{bmatrix} , \\ \boldsymbol{b} &= \begin{bmatrix} 0.11 & -0.21 & 0.28 & -0.21 & 0.11 \end{bmatrix} , \\ \boldsymbol{c} &= \begin{bmatrix} 0 & 1 & 5 & -0.5 \end{bmatrix} . \end{aligned}$$

The two systems are depicted in Fig. 7.1. The variance of the output noise is obtained by matching different signal-to-noise ratios (SNR). In particular, we test the proposed methods for SNR = 10, 20, 40 dB. Thus, we have in total 6 experimental conditions. For each experimental condition we generate 100 Monte Carlo, each with a different noise realization. The performance of the methods is evaluated by assessing the accuracy in tracking the output of a noiseless test set $y_{t,\text{test}}$ of length $N = 500$, obtained by feeding the systems with a i.i.d. Gaussian sequence. We use the normalized mean absolute error (%MAE) (i.e. see Appendix D) and the normalized mean square error (%MSE), defined as

$$\%\text{MSE} = 100 \frac{\sqrt{\sum_{t=1}^{N} \left(y_{\text{test}}(t) - \hat{y}_{\text{test}}(t)\right)^2}}{\sqrt{\sum_{t=1}^{N} \left(y_{\text{test}}(t) - \text{mean}(y_{\text{test}}(t))\right)^2}} . \tag{7.29}$$

7.4.1 Experiments using the parametric and the polynomial approaches

First, we test the methods presented in Section 7.3.1 and in Section 7.3.2. We refer to the two methods as **2-E-P** (two-experiment parametric) and **2-E-Poly** (two-experiment

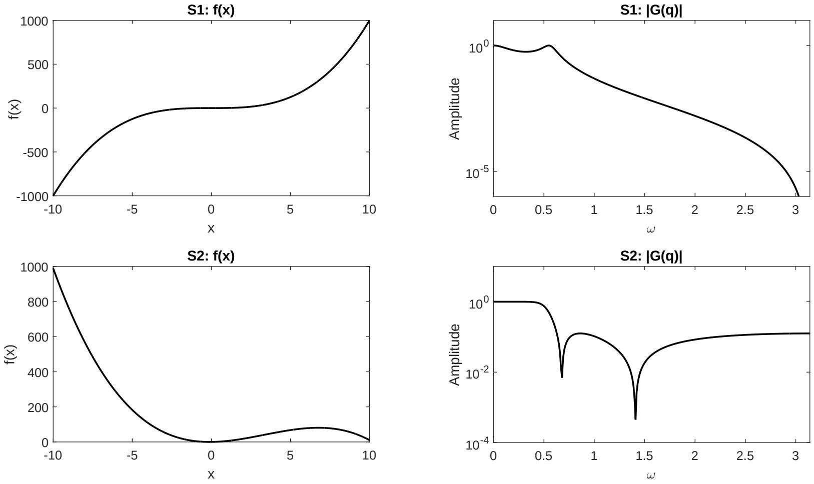


Figure 7.1: Blocks composing S1 and S2. (Up) S1. (Down) S2. (Left) Nonlinear blocks. (Right) Magnitude of the transfer functions of the LTI blocks.

polynomial). We use the same input signals for both methods. In the first stage of the approach, we feed the system with the signal $u_{(1)}(t) = \sin(0.05t)$, keeping $N_1 = 500$ transient-free samples. In the second stage, we use random white Gaussian noise with unit variance as input $u_{(2)}(t)$, collecting $N_2 = 500$ samples. The two methods are compared with a single-stage PEM-based estimator, implemented through the Matlab command `nlhw` (see Ljung, Zhang, Lindskog, Iouditski, and Singh (2007) for details); we refer to this method as **NLHW**. To get a fair comparison, the data used by **NLHW** are obtained by feeding the system with a random white Gaussian sequence of length $N_1 + N_2$ with variance equal to the variance of the sequence $[u_{(1),1} \; \cdots \; u_{(1),N_1} \; u_{(2),1} \; \cdots \; u_{(2),N_2}]$.

In Table 7.1 we report the medians of the results obtained in the 6 experiments. As can be seen, on S1 the performance of the three methods are comparable to each other, while **NLHW** fails in providing a good model of S2. The motivation is that the method often falls into a local minimum of the cost function related to the PEM optimization problem. Separating the identification of the LTI block from the static nonlinearity avoids this issue; thus, the proposed methods **2-E-P** and **2-E-Poly** give reliable results for both S1 and S2.

Table 7.1: Median values of %MAE and %MSE over 100 Monte Carlo runs obtained by the three parametric methods on the 6 tested experimental conditions.

SNR (dB)	**Method**	**S1**		**S2**	
		%MAE	%MSE	%MAE	%MSE
10	**2-E-P**	0.19	4.82	0.5	6.97
	2-E-Poly	0.2	5.1	0.49	6.87
	NLHW	0.19	5.28	3.3	39.63
20	**2-E-P**	0.15	4.13	0.34	4.74
	2-E-Poly	0.16	4.07	0.33	4.67
	NLHW	0.16	3.71	3.72	51.12
40	**2-E-P**	0.14	3.44	0.25	3.55
	2-E-Poly	0.13	3.33	0.25	3.49
	NLHW	0.09	2.25	3.61	44.35

7.4.2 Experiments using the nonparametric approach

For the nonparametric approach, the input used in the first stage is $u(t) = 10\sin(t + \pi/2)$. We collect $N_1 = 500$ steady-state output samples. In the second stage the systems are excited with an uniformly distributed random signal, collecting $N_2 = 500$ samples.

In the proposed nonparametric approach, dubbed **2-E-NP**, $\hat{f}(\cdot)$ is estimated using the LS-SVM approach described in Section 7.3.3, under a cross validation scheme. To solve the optimization problem associated with hyperparameter selection, we use coupled simulated annealing (Xavier-de Souza et al., 2009) and simplex (Nelder & Mead, 1965); an implementation of these techniques is available in the toolbox LS-SVMlab v1.8[1]. The results of the Monte Carlo simulations are displayed through the box plots of Fig. 7.2.

The proposed method is compared with the method introduced in Chapter 4, where the Best Linear Approximation (BLA) (Pintelon & Schoukens, 2012) is used together with LS-SVM to model the linear and nonlinear blocks, respectively. This method is referred to as **BLA+LS-SVM**. This method uses 10 random phase multisine (see Pintelon and Schoukens (2012)) inputs $u_{\text{BLA}}^{(i)}(t)$ with their corresponding outputs $y_{\text{BLA}}^{(i)}(t)$. Here $t = 1, \ldots, 2500$ and $i = 1, \ldots, 10$. Using those signals, the BLA is calculated and as an approximation to the LTI block. Using a second data set with a new input signal $u_{nl}(t)$, the intermediate variable $\hat{x}_{nl}(t)$ is estimated. With $\hat{x}_{nl}(t)$ and the known output $y_{nl}(t)$ a model of the nonlinear block is estimated using LS-SVM. In this case, $u_{nl}(t)$

[1] http://www.esat.kuleuven.be/sista/lssvmlab/

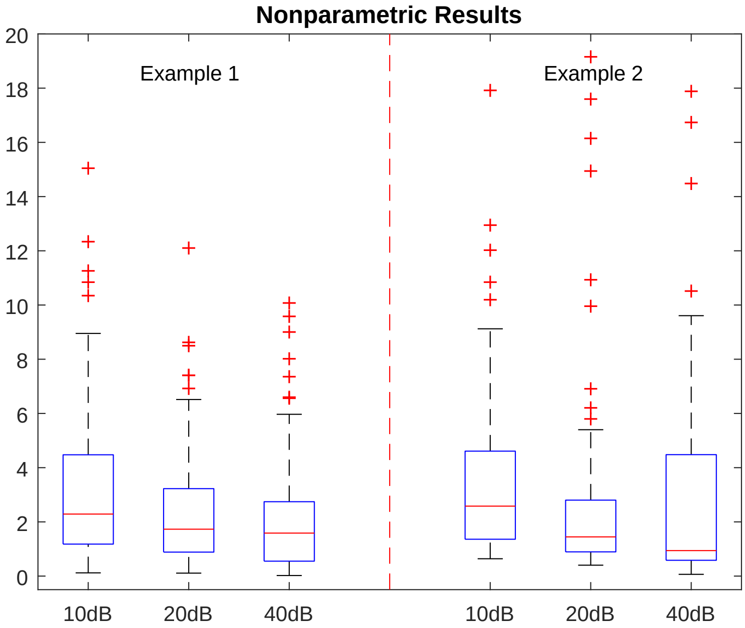


Figure 7.2: Box plots of the obtained $\%$MAE for the 100 Monte Carlo runs of the 6 experiments.

is the same sinusoidal signal described used by **2-E-NP**. To train the LS-SVM model, only the last 500 samples of $\hat{x}_{nl}(t)$ and $y_{nl}(t)$ are used.

In Table 7.2 we report the median values of $\%$MAE obtained using the two nonparametric methods. It can be seen that the proposed technique outperforms **BLA+LS-SVM**, despite the simplicity of the input employed in the first stage of the procedure, as compared with the random phase multisine signal required by BLA (note also that **BLA+LS-SVM** uses a 5 times longer input in this stage). It should be noted that the performance of **2-E-NP** is lower than those obtained by the parametric methods. This can be explained by the fact that the parametric approaches exploit more detailed prior information on the static nonlinearity, because it is known that the nonlinearity is a third-order polynomial, while when **2-E-NP** is used, it is only known that the nonlinearity is smooth.

Table 7.2: Median values of %MAE over 100 Monte Carlo runs obtained by the two nonparametric methods on the 6 tested experimental conditions.

SNR (dB)	**Method**	**S1** %NMAE	**S2** %MAE
10	**2-E-NP**	2.29	2.58
	BLA+LS-SVM	16.7	53.81
20	**2-E-NP**	1.73	1.45
	BLA+LS-SVM	29.08	4.68
40	**2-E-NP**	1.59	0.94
	BLA+LS-SVM	13.15	4.72

7.5 Conclusions

We have proposed in this chapter a new method for Wiener system identification. Remarkably, it is shown that for Wiener systems, a poorly exciting signal –such as a sinusoid– can help estimating (part of) the system by means of relatively simple least-squares based procedures. The main idea underlying the method is that we can separate the estimation of the static nonlinear function from the identificaiton of the LTI block composing the Wiener system. To do so, we have to excite the system using two different inputs. The first input is a sinusoid, which, after the transient effect of the LTI system has vanished, permits to estimate the static nonlinearity as a function of the phase delay introduced by the LTI block. We have described three different approaches to nonlinearity estimation, namely a parametric, a polynomial, and a nonparametric approach. Depending on the adopted approach, the phase delay is also estimated so that the static nonlinearity is recovered. Using the information on the static nonlinearity, we use a persistently exciting input to identify the LTI block. The proposed method is shown to compare favorably with other techniques for Wiener system identification.

Future challenges are to extend the two-experiment approach to more involved model structures, such as Wiener-Hammerstein and Hammerstein-Wiener systems.

Part III

Impulse Response System Identification

In this part, two newly developed methods for system identification of Hammerstein systems are presented. The proposed methodologies not only take into account the information about the structure of the system, but also exploit the structural information of the underlying systems so that the impulse response of the associated linear systems can be approximated.

In Chapter 8 SISO Hammerstein systems are considered based on Castro-Garcia, Agudelo, and Suykens (2017b). That work is extended to include the MIMO Hammerstein systems In Chapter 9 based on Castro-Garcia, Agudelo, and Suykens (2017a). In both scenarios the respective method consists of two stages: First, the impulse response of the system is approximated. In the SISO case this can be done straightforwardly, but in the MIMO one it becomes a more involved process. Afterward, using the found estimation of the linear block, LS-SVM is used to model the nonlinearity. These methods have as their main advantage their versatility with respect to the class of systems that can be modeled.

Chapter 8

Impulse Response Constrained LS-SVM modeling for Hammerstein System Identification

8.1 Introduction

The proposed methodology in this chapter not only takes into account the information about the structure of the system, but also exploits the structural information so that the impulse response of the linear system can be approximated.

The method proposed can be separated in two stages: First the system's impulse response is estimated. Then, using this, an LS-SVM model of the whole system is estimated (i.e. as opposed to modeling its component blocks). Even though at the second stage a model for the whole system is obtained, it can still be separated into the corresponding models of the linear block and the nonlinearity.

The capabilities of the method will be illustrated through several Monte Carlo simulations covering two examples and it will be shown how the measurement

This chapter is based on:
Castro-Garcia, R., Agudelo, O. M., Suykens, J. A. K. (2017b). *Impulse Response Constrained LS-SVM modeling for Hammerstein System Identification*. In proceedings of the 20th world congress of the International Federation of Automatic Control (IFAC 2017), Toulouse, France. (pp. 14611 – 14616).

noise (white Gaussian noise with zero mean) affects the behavior of the proposed methodology.

Given that a modified formulation of LS-SVM is used in order to include the estimated Impulse Response, two different methods for tuning the parameters are also discussed. These methods are Genetic Algorithms and Simulated Annealing for global optimization on a validation set.

This chapter is organized as follows: In Section 8.2 a quick review of the impulse response of Hammerstein systems is presented. In Section 8.3 the proposed method is presented. Section 8.4 shows the results found when applying the described methodology on two simulation examples. Finally, in Section 8.5, the conclusions are presented.

8.2 Hammerstein Impulse Response

Throughout this chapter, we use the discrete time framework. Given this, we define an impulse as a Kronecker delta function. This means for $t \in \mathbb{N}$:

$$u_{imp}(t) = u_i\delta(t) = \begin{cases} u_i & \text{for } t = 0 \\ 0 & \text{for } t \neq 0. \end{cases} \tag{8.1}$$

This representation shows that the $\delta(t)$ function, by definition a unit impulse, is rescaled by a factor u_i.

In order to obtain an impulse response matrix from a Hammerstein system, it is enough to apply such an impulse as input and measure the corresponding output. This can be easily understood if we consider that the first block contains a static nonlinearity and therefore, the resulting intermediate variable $x_{imp}(t)$ for the impulse input $u_{imp}(t)$ is a rescaled version of $u_{imp}(t)$. The initial value is simply the value of the impulse multiplied by an unknown constant η, that is:

$$x_{imp}(t) = \begin{cases} \eta u_i & \text{for } t = 0 \text{ with } \eta \neq 0 \\ 0 & \text{for } t \neq 0. \end{cases} \tag{8.2}$$

The linear part will be excited then by $x_{imp}(t)$ and the corresponding output $y_{imp}(t)$ can be used to construct an Impulse Response Matrix $\boldsymbol{M}_{IR}$ (Ljung, 1999):

$$\boldsymbol{M}_{IR} = \begin{bmatrix} y_{imp}(0) & 0 & 0 & \cdots & 0 \\ y_{imp}(1) & y_{imp}(0) & 0 & \cdots & 0 \\ y_{imp}(2) & y_{imp}(1) & y_{imp}(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{imp}(N-1) & y_{imp}(N-2) & y_{imp}(N-3) & \cdots & y_{imp}(0) \end{bmatrix}. \tag{8.3}$$

This is very convenient as we can easily obtain a rescaled version of the impulse response of the system. Note however that this measured impulse response will contain noise.

The rescaling of the impulse response does not represent a problem as the LS-SVM model will take care of it in the next stage. In other words, the rescaling of the approximated linear block has no effect on the input-output behavior of the Hammerstein model (i.e. any pair of $\{f(u(t))/\eta,\ \eta G(q)\}$ with $\eta \neq 0$ would yield identical input and output measurements), as will be shown in Section 8.3.1.

It is important to highlight that the impulse excitation has the disadvantage of not being persistently exciting. In practice the amplitude of such signals is limited and hence, within the available experiment time, more information can be collected by using richer excitations. In this sense it is possible to use for instance a Pseudo Random Binary Signal (PRBS) input $u_{pr}(t)$ switching between zero and a non zero constant $\bar{u}$. This means that

$$\begin{aligned} y_{pr}(t) &= \boldsymbol{M}_{IR} x_{pr}(t) \\ &= \eta \boldsymbol{M}_{IR} u_{pr}(t), \end{aligned} \tag{8.4}$$

with $f(\bar{u}) = \eta\bar{u}$ and therefore $\hat{\boldsymbol{M}}_{IR} = \eta \boldsymbol{M}_{IR}$ can be estimated from the known $u_{pr}(t)$ and $y_{pr}(t)$.

8.3 Proposed Method

8.3.1 Impulse Response Constrained LS-SVM

The proposed method aims to integrate the Impulse Response Matrix $\boldsymbol{M}_{IR}$ of the Hammerstein system as defined in Section 8.2 into the LS-SVM formulation presented in Section 1.7.1. To do this, the constrained optimization problem is reformulated for any input/output data as follows:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e}} \quad & \tfrac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \tfrac{\gamma}{2}\boldsymbol{e}^\top \boldsymbol{e} \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^\top \boldsymbol{w} + \boldsymbol{1}_N b) + \boldsymbol{e}. \end{aligned} \tag{8.5}$$

Here, $\boldsymbol{w} \in \mathbb{R}^{n_h}$, the input matrix is $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]^\top$ and the elements of the input signal $\boldsymbol{u}_i \in \mathbb{R}^n$. Also $\boldsymbol{y}, \boldsymbol{e}, \boldsymbol{1}_N \in \mathbb{R}^N$ with $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]$ the output, $\boldsymbol{e} = [e_1, e_2, \ldots, e_N]$ the errors and $\boldsymbol{1}_N$ a vector of ones. Finally, $\boldsymbol{\Phi}_{\boldsymbol{U}} \in \mathbb{R}^{n_h \times N}$, or equivalently:

$$\boldsymbol{\Phi}_{\boldsymbol{U}} = [\varphi(\boldsymbol{u}_1), \varphi(\boldsymbol{u}_2), \ldots, \varphi(\boldsymbol{u}_N)] \tag{8.6}$$

with $\varphi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ the feature map to a high dimensional (possibly infinite) space. Also, we have that $\boldsymbol{\Omega} = \boldsymbol{\Phi}_{\boldsymbol{U}}^\top \boldsymbol{\Phi}_{\boldsymbol{U}}$.

Note that in the constraint of (8.5) the separation of the blocks is clear: The Impulse Response Matrix $\boldsymbol{M}_{IR}$ models the linear block and is multiplied with the output of the nonlinear model given by $\boldsymbol{\Phi}_{\boldsymbol{U}}^{\top}\boldsymbol{w} + \mathbf{1}_N b$. However, note also that we only consider one global error for the whole model. This means that the tuning of the Impulse Response Constrained LS-SVM eventually will have to deal with the errors of the two blocks, i.e. the errors introduced by the Impulse Response Matrix and the errors introduced by the selection of the parameters in the nonlinear block.

From the Lagrangian:

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{e}; \boldsymbol{\alpha}) \quad = \quad \tfrac{1}{2}\boldsymbol{w}^{\top}\boldsymbol{w} + \gamma\tfrac{1}{2}\boldsymbol{e}^{\top}\boldsymbol{e} - \boldsymbol{\alpha}^{\top}(\boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^{\top}\boldsymbol{w} + \mathbf{1}_N b) + \boldsymbol{e} - \boldsymbol{y}), \tag{8.7}$$

the optimality conditions are then derived:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} & = 0 \quad \rightarrow \boldsymbol{w} = \boldsymbol{\Phi}_{\boldsymbol{U}}\boldsymbol{M}_{IR}^{\top}\boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial b} & = 0 \quad \rightarrow 0 = \mathbf{1}_N^{\top}\boldsymbol{M}_{IR}^{\top}\boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial e_i} & = 0 \quad \rightarrow \boldsymbol{e} = \boldsymbol{\alpha}/\gamma \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} & = 0 \quad \rightarrow \boldsymbol{y} = \boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^{\top}\boldsymbol{w} + \mathbf{1}_N b) + \boldsymbol{e}. \end{cases} \tag{8.8}$$

By elimination of $\boldsymbol{w}$ and $\boldsymbol{e}$, the last equation can be rewritten as:

$$\boldsymbol{y} = \boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^{\top}\boldsymbol{\Phi}_{\boldsymbol{U}}\boldsymbol{M}_{IR}^{\top}\boldsymbol{\alpha} + \mathbf{1}_N b) + \frac{\boldsymbol{\alpha}}{\gamma} \tag{8.9}$$

and the following linear system is obtained:

$$\left[\begin{array}{cc} 0 & \mathbf{1}_N^{T}\boldsymbol{M}_{IR}^{\top} \\ \boldsymbol{M}_{IR}\mathbf{1}_N & \boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^{\top} + \frac{1}{\gamma}\boldsymbol{I}_N \end{array}\right]\left[\begin{array}{c} b \\ \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \boldsymbol{y} \end{array}\right]. \tag{8.10}$$

For a new input signal $\boldsymbol{U}_d \in \mathbb{R}^{n\times D}$ with elements $\boldsymbol{d} \in \mathbb{R}^n$ and the training input $\boldsymbol{U} \in \mathbb{R}^{n\times N}$ with elements $\boldsymbol{u} \in \mathbb{R}^n$, let us define a matrix $\boldsymbol{K} \in \mathbb{R}^{D\times N}$ whose entries are defined as

$$\boldsymbol{K}_{i,j} = \exp\left(\frac{-\|\boldsymbol{u}_j - \boldsymbol{d}_i\|_2^2}{\sigma^2}\right), \tag{8.11}$$

with $i = 1, \ldots, D$ and $j = 1, \ldots, N$. Note that in the case where $\boldsymbol{U}_d = \boldsymbol{U}$ then $\boldsymbol{K} = \boldsymbol{\Omega}$.

If $N \neq D$ we also need to define an additional matrix $\boldsymbol{M}_{New}$. This matrix will be a re-sized version of $\boldsymbol{M}_{IR}$ in order to make it coincide with the new data set (i.e.

$\boldsymbol{M}_{New} \in \mathbb{R}^{D \times D}$). Note that if the new data set is longer than the training one, and assuming that the impulse response $\boldsymbol{y}_{imp}$ is long enough to allow the system to settle down, $\boldsymbol{M}_{New}$ can be generated by extending $\boldsymbol{y}_{imp}$ with zeros. On the other hand, if the new data set is shorter than the training one, $\boldsymbol{M}_{New}$ can be generated by truncating $\boldsymbol{y}_{imp}$. Of course, if $N = D$ then $\boldsymbol{M}_{IR} = \boldsymbol{M}_{New}$.

Finally, we can define the estimated output for $\boldsymbol{U}_d$ as:

$$\hat{\boldsymbol{y}}(\boldsymbol{U}_d) = \boldsymbol{M}_{New}\boldsymbol{K}\boldsymbol{M}_{IR}^\top\boldsymbol{\alpha} + \boldsymbol{M}_{New}\mathbf{1}_N b. \tag{8.12}$$

In this final formulation, the clear separation between the linear and nonlinear blocks present in (8.5) is lost. However, it is still possible to make a separation between the two blocks by factorizing $\boldsymbol{M}_{New}$. This leads then to

$$\hat{\boldsymbol{y}}(\boldsymbol{U}_d) = \boldsymbol{M}_{New}(\boldsymbol{K}\boldsymbol{M}_{IR}^\top\boldsymbol{\alpha} + \mathbf{1}_N b). \tag{8.13}$$

In Section 8.4 it will be illustrated how from (8.13) we can recover a good approximation to the nonlinearity.

8.3.2 Role of Regularization

It is important to highlight the importance of the regularization in the found model.

As shown in (8.10), $\boldsymbol{y}$ can be expressed as:

$$\boldsymbol{y} = \boldsymbol{M}_{IR}\mathbf{1}_N b + \boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top\boldsymbol{\alpha} + \frac{\boldsymbol{I}_N}{\gamma}\boldsymbol{\alpha}. \tag{8.14}$$

If we were to calculate the output of the found model for the input of the training data set, we would have then:

$$\tilde{\boldsymbol{y}} \quad = \quad \boldsymbol{M}_{IR}\mathbf{1}_N b + \boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top\boldsymbol{\alpha} \tag{8.15}$$

It is clear then from (8.14) and (8.15) that $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\alpha}/\gamma$, this leads to

$$\tilde{\boldsymbol{y}} = \boldsymbol{y} - \frac{1}{\gamma}\left(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top + \frac{\boldsymbol{I}_N}{\gamma}\right)^{-1}(\boldsymbol{y} - \boldsymbol{M}_{IR}\mathbf{1}_N b) \tag{8.16}$$

Now, let us assume that a change $\boldsymbol{\Delta}_v$ in the measurement noise occurs and let us analyze the effect it has in $\tilde{\boldsymbol{y}}$:

$$\begin{aligned}\tilde{\boldsymbol{y}} + \boldsymbol{\Delta}_{\tilde{y}} \quad &= \quad \boldsymbol{y} + \boldsymbol{\Delta}_v - \tfrac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top + \tfrac{\boldsymbol{I}_N}{\gamma})^{-1} \\ &\qquad (\boldsymbol{y} + \boldsymbol{\Delta}_v - \boldsymbol{M}_{IR}\mathbf{1}_N b) \\ &= \quad \boldsymbol{y} - \tfrac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top + \tfrac{\boldsymbol{I}_N}{\gamma})^{-1} \\ &\qquad (\boldsymbol{y} - \boldsymbol{M}_{IR}\mathbf{1}_N b) + \boldsymbol{\Delta}_v \\ &\qquad -\tfrac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top + \tfrac{\boldsymbol{I}_N}{\gamma})^{-1}\boldsymbol{\Delta}_v.\end{aligned} \tag{8.17}$$

Therefore

$$\begin{aligned} \boldsymbol{\Delta}_{\tilde{y}} &= \boldsymbol{\Delta}_v - \tfrac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top + \tfrac{\boldsymbol{I}_N}{\gamma})^{-1}\boldsymbol{\Delta}_v \\ &= (\boldsymbol{I} - \tfrac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top + \tfrac{\boldsymbol{I}_N}{\gamma})^{-1})\boldsymbol{\Delta}_v. \end{aligned} \tag{8.18}$$

From (8.18) it is evident that the effect of $\boldsymbol{\Delta}_v$ in $\tilde{\boldsymbol{y}}$ is heavily dependent on γ. Let us now assume that $\gamma \to 0$:

$$\boldsymbol{\Delta}_{\tilde{y}} \approx \left(\boldsymbol{I} - \frac{1}{\gamma}\left(\frac{\boldsymbol{I}_N}{\gamma}\right)^{-1}\right)\boldsymbol{\Delta}_v = 0. \tag{8.19}$$

Here, $\boldsymbol{\Delta}_v$ has no effect over $\tilde{\boldsymbol{y}}$. This result was to be expected as the errors considered in (8.5) have no impact at all given that γ is so small.

Let us assume now that $\gamma \to \infty$ (i.e. the errors in (8.5) are given a very high weigth):

$$\boldsymbol{\Delta}_{\tilde{y}} \approx (\boldsymbol{I} - \frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^\top)^{-1})\boldsymbol{\Delta}_v = \boldsymbol{\Delta}_v. \tag{8.20}$$

Here the model would follow the training points perfectly regardless of the noise. This is an undesirable effect as well as it clearly leads to overfitting.

8.3.3 Method Summary

In Fig. 8.1 the algorithm of the proposed method is summarized. Note that the elements in the input signals used are scalars and therefore, we drop the matrix notation.

8.4 Simulation Results

The proposed methodology was applied to two systems in the discrete time framework. In order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D. The first system (i.e. Example 1) was generated through a nonlinear block:

$$x(t) = u(t)^3 \tag{8.21}$$

and a linear block:

$$y(t) = \frac{q^6+0.8q^5+0.3q^4+0.4q^3}{q^6-2.789q^5+4.591q^4-5.229q^3+4.392q^2-2.553q+0.8679}x(t). \tag{8.22}$$

The second system (i.e. Example 2) was generated through a nonlinear block:

$$x(t) = -0.5u^3 + 5u^2 + u \tag{8.23}$$

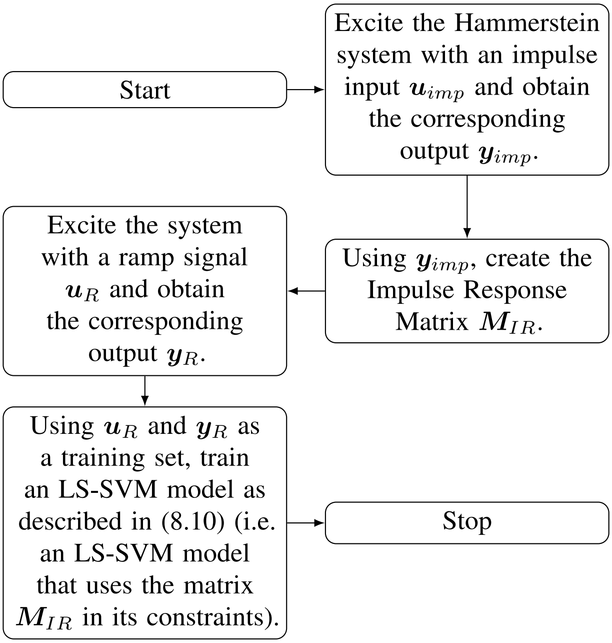

Start
Excite the Hammerstein system with an impulse input $\boldsymbol{u}_{imp}$ and obtain the corresponding output $\boldsymbol{y}_{imp}$.
Using $\boldsymbol{y}_{imp}$, create the Impulse Response Matrix $\boldsymbol{M}_{IR}$.
Excite the system with a ramp signal $\boldsymbol{u}_R$ and obtain the corresponding output $\boldsymbol{y}_R$.
Using $\boldsymbol{u}_R$ and $\boldsymbol{y}_R$ as a training set, train an LS-SVM model as described in (8.10) (i.e. an LS-SVM model that uses the matrix $\boldsymbol{M}_{IR}$ in its constraints).
Stop

Figure 8.1: Summary of the method showing the main steps.

and a linear block:

$$y(t) = \frac{0.004728q^3 + 0.01418q^2 + 0.01418q + 0.004728}{q^3 - 2.458q^2 + 2.262q - 0.7654} x(t). \tag{8.24}$$

The examples can be visualized in Fig. 8.2.

Both systems were initially excited using an impulse signal $\boldsymbol{u}_{imp}$ (i.e. $\boldsymbol{u}_{imp} = [10, 0, \ldots, 0]^\top$) and the corresponding $\boldsymbol{y}_{imp}$ were retrieved. The Impulse Response Matrices $\boldsymbol{M}_{IR}$ were created using the corresponding $\boldsymbol{y}_{imp}$. Next, a ramp like signal is used to excite the systems (i.e. $\boldsymbol{u}_R$) and their corresponding outputs $\boldsymbol{y}_R$ are retrieved. Finally, using $\boldsymbol{u}_R$, $\boldsymbol{y}_R$ and $\boldsymbol{M}_{IR}$, models for the systems are estimated as explained in Section 8.3.

The estimated models were tested in an independent test set. The inputs of this set $\boldsymbol{u}_T$ are Multilevel Pseudo Random Signals with 2% switching probability and amplitude values drawn from a uniform distribution with amplitudes in the interval $[-10, 10]$. All the signals in the presented examples consist of 500 samples.

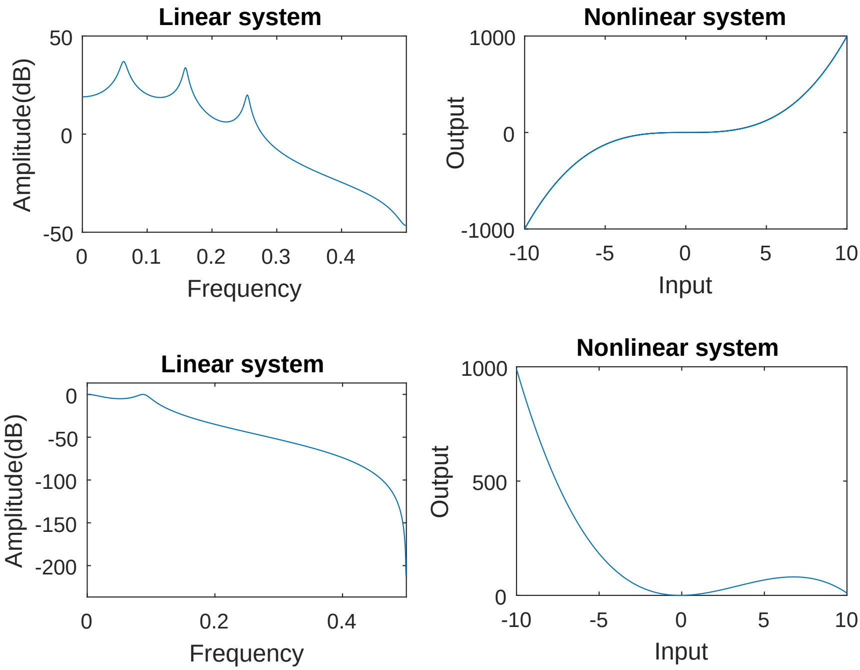
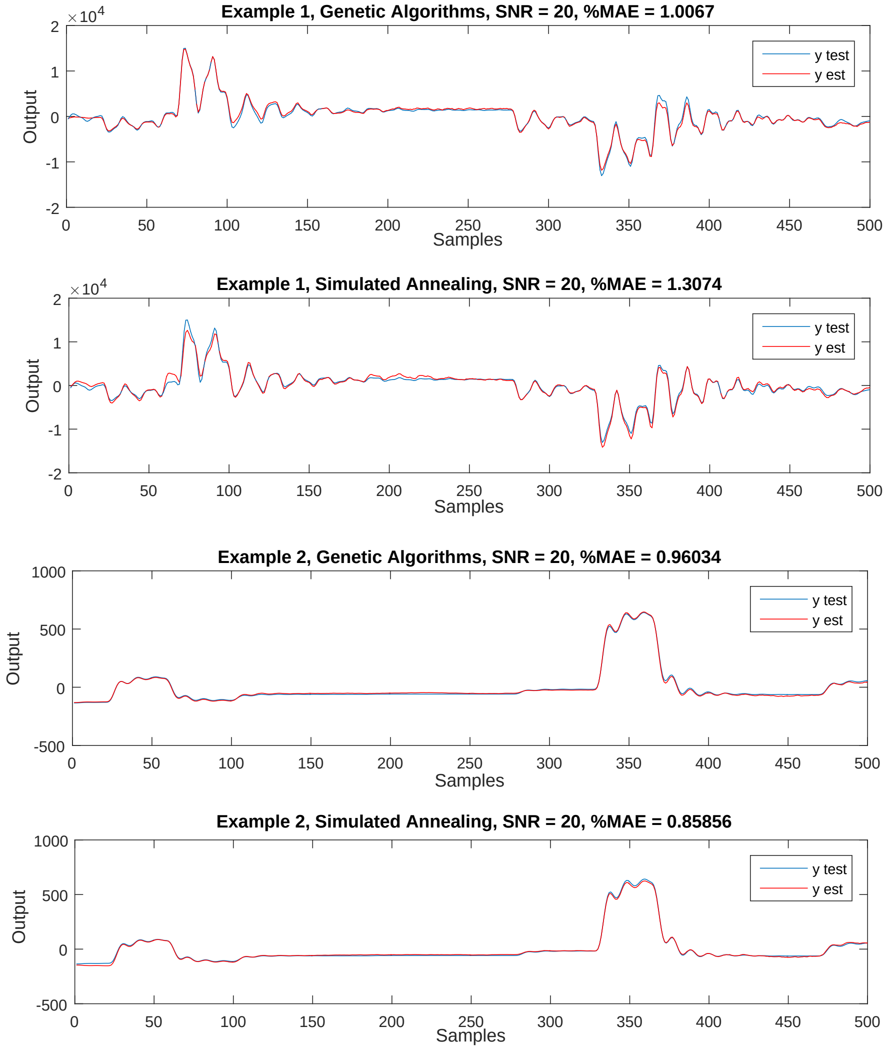


Figure 8.2: Example 1: (Left) Linear block in the frequency domain (normalized). (Rigth) Nonlinear block. (Top) Example 1. (Bottom) Example 2.

Two different methods for tuning the hyper-parameters (i.e. σ and γ) were tried, namely Genetic Algorithms and Simulated Annealing. These methods were used through validation sets. Results for Examples 1 and 2 are shown in Fig. 8.3 for the different tuning methods.

White Gaussian noise with zero mean was added to the output of the systems such that a resulting Signal to Noise Ratio (SNR) of 20 dB was obtained. It is clear that even in the presence of noise, the proposed methodology works very well when Simulated Annealing or Genetic Algorithms are used.

In Fig. 8.4, the found nonlinearities for Examples 1 and 2 are depicted. These estimations were done following the separation between the linear and nonlinear blocks in the found model explained in (8.13). This is, $\hat{\boldsymbol{x}} = \boldsymbol{K}\boldsymbol{M}_{IR}^{\top}\boldsymbol{\alpha} + \mathbf{1}_N b$ with a $\boldsymbol{K}$ matrix generated as explained in Section 8.3.1 using the input of the training data $\boldsymbol{U}$ and an input $\boldsymbol{U}_{NL} = [-10, -9, \ldots, 9, 10]$.



Example 1, Genetic Algorithms, SNR = 20, %MAE = 1.0067

Example 1, Simulated Annealing, SNR = 20, %MAE = 1.3074

Example 2, Genetic Algorithms, SNR = 20, %MAE = 0.96034

Example 2, Simulated Annealing, SNR = 20, %MAE = 0.85856

Figure 8.3: Results for Examples 1 and 2.

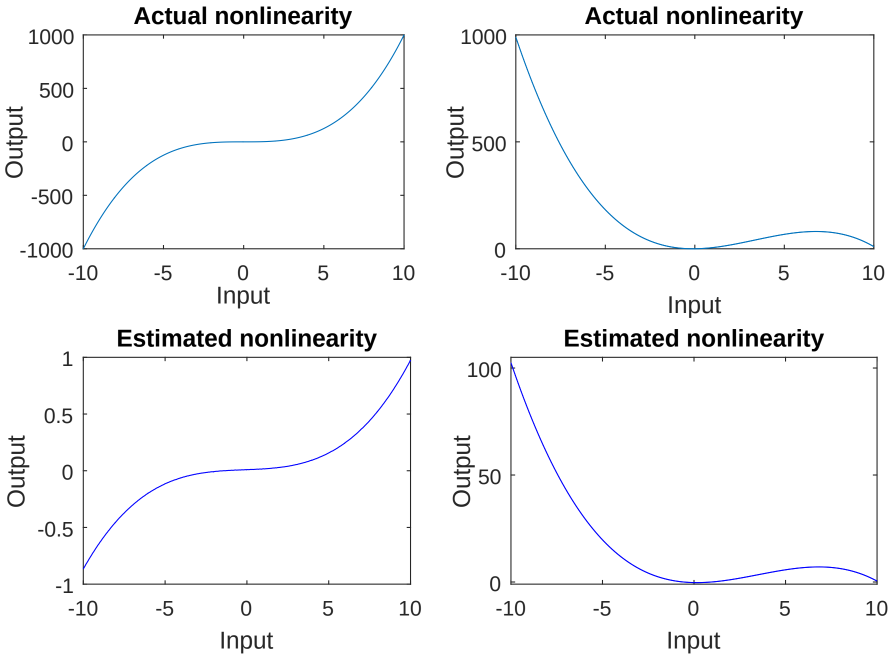


Figure 8.4: (Left) Nonlinearity of Example 1. (Rigth) Nonlinearity of Example 2. (Top) Actual Nonlinearities. (Bottom) Estimated nonlinearities. Results corresponding to the examples of Figs. 8.3.

As can be seen, even though the scales are different, the shapes of the estimated nonlinearities are very similar to the actual ones. Again, it is important to note that this scaling factor is of no consequence in the input-output behavior as any pair of $\{f(u(t))/\eta,\ \eta G(q)\}$ with $\eta \neq 0$ would yield identical input and output measurements.

In addition, in order to show the effect of parameter tuning during the modeling of the system Figs. 8.5 and 8.6 are presented. There σ and γ are alternatively fixed while the other varies in a wide range. The corresponding normalized mean absolute errors (%MAE) are displayed for the training and test set of Examples 1 and 2 for Genetic Algorithms and Simulated Annealing correspondingly.

Fig. 8.7 summarizes the result of 100 Monte Carlo simulations for each example and each tuning methodology.

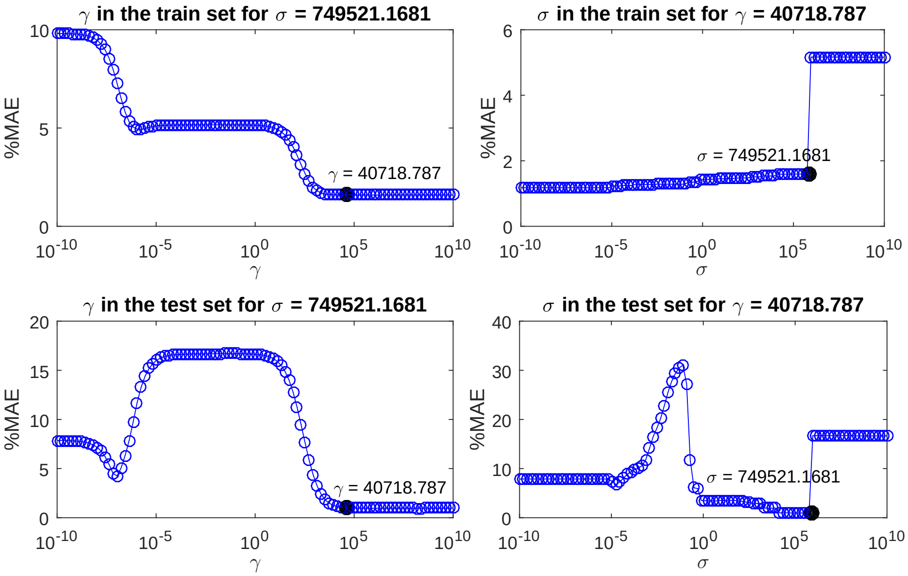


Figure 8.5: Example1. Behavior of the error with respect to γ (left) and σ (right). (Top) Training set results. (Bottom) Test set results. The black dot shows the selected value.

From these results it can be seen that both Genetic Algorithms and Simulated Annealing can achieve very good results even in the presence of noise. However, it is also clear that the levels of noise used lead the results obtained with Genetic Algorithms to be slightly less homogeneous.

The proposed method takes the underlying structure of the system into account through the modified constraint in (8.5), therefore it is expected to produce better models than those obtained with purely black box methods like the NARX LS-SVM discussed in Suykens et al. (2002). Table 8.1 shows the results of the comparison between the proposed method (i.e. IR+LS-SVM) and a NARX LS-SVM with 10 lags input and 10 lags output in a test set. Additionally, the method is compared against the MathWorks' System Identification Toolbox (SITB) (Ljung et al., 2007). Both, the single hidden layer neural network with sigmoid neurons (i.e. `SigmoidNet`) and the piecewise linear estimator (i.e. `PWlinear`) are considered.

For the NARX LS-SVM, a ramp signal $\boldsymbol{u}_R$ is used for training and a 10-fold cross-validation scheme was used with a combination of Coupled Simulated Annealing (Xavier-de Souza et al., 2009) and simplex search for the tuning of the

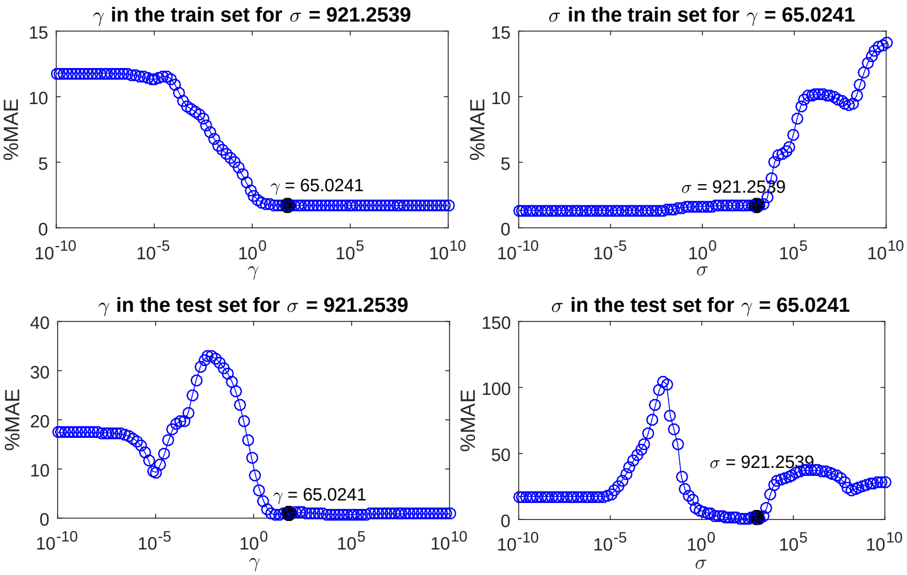


Figure 8.6: Example 2. Behavior of the error with respect to γ (left) and σ (right). (Top) Training set results. (Bottom) Test set results. The black dot shows the selected value.

hyper-parameters (i.e. LS-SVMlab v1.8[1]). For the `SigmoidNet` 25 neurons are used. Similarly, for the `PWlinear` 25 points are used for the nonlinear modeling. In both cases the order of the linearity is chosen by observation of the behavior of the noiseless case. This means that the order of numerator and denominator are the same in both examples and both methodologies, this is 6 and 3 for Examples 1 and 2 respectively.

To train the SITB methods, a 500 points ramp signal ranging from -15 to 15 was created. This signal was randomly shuffled so the resulting training signal is rich in its frequency content while covering all the input range.

It can be seen that the proposed method clearly outperforms the purely black box approach of NARX LS-SVM. Also, when compared with the SITB methods the results of the proposed method are in general better. Note that the order of the linear block was manually picked for the SITB methods in a noiseless environment while for the proposed method the process is fully automated.

[1] http://www.esat.kuleuven.be/stadius/lssvmlab/

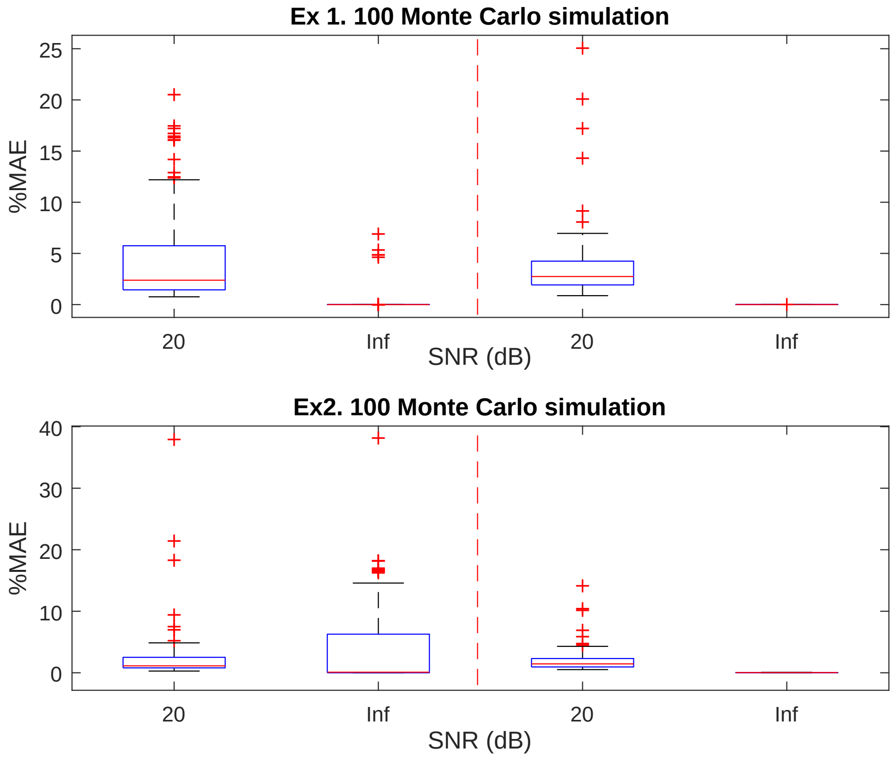


Figure 8.7: Results of 100 Monte Carlo simulations using (Left) Genetic Algorithms and (Rigth) Simulated Annealing with 20dB SNR and no noise for Example 1 (Top) and Example 2 (Bottom).

8.5 Conclusions

The proposed method in this chapter includes information about the structure of the Hammerstein system within an LS-SVM formulation. Also, we exploit the structure of the system for obtaining a rescaled impulse response and the fact that such a rescaling is not a problem for the modeling of the system as a whole.

The results indicate that when the structure of the system is taken into account, a substantial improvement can be achieved in the resulting modeling. Also, they show that the method is effective in the presence of zero mean white Gaussian noise.

Table 8.1: $\%MAE$ Comparison. Median values are offered for 100 Monte Carlo simulations for each case.

	SNR 20dB		No Noise	
Method	Ex 1	Ex 2	Ex 1	Ex 2
NARX LS-SVM	2.8154	2.0174	0.5668	0.4592
IR+LS-SVM (SA)	2.742	1.4452	0.0048	$\mathbf{0.03433}$
IR+LS-SVM (GA)	2.3843	$\mathbf{1.1288}$	$\mathbf{8.6216 \times 10^{-5}}$	0.0905
SITB PWLinear	$\mathbf{1.9487}$	4.1317	0.2559	0.447
SITB SigmoidNet	3.3196	6.8528	0.3486	0.1992

For this method, the kernel parameter σ and the regularization parameter γ have to be tuned. To this end, two techniques were used and compared using Monte Carlo simulations.

It is interesting to note that in the initial formulation, a clear separation in the modeling of the linear and nonlinear blocks is present. However, when the final model to be used is derived from the dual formulation, that separation is no longer clear anymore.

The solution of the model follows from solving a linear system of equations. This is a clear advantage over other methodologies like the overparametrization presented in Goethals et al. (2005).

Chapter 9

Impulse Response Constrained LS-SVM modeling for MIMO Hammerstein System Identification

9.1 Introduction

Despite the abundance of the existing literature on Hammerstein system identification, the vast majority focuses on Single-Input Single-Output (SISO) case. For the Multiple-Input Multiple-Output (MIMO) case, however, much less works exist. Examples of methods for the identification of MIMO Hammerstein systems include for instance: Gomez and Baeyens (2004), Jeng and Huang (2008), Goethals et al. (2005), Lee et al. (2004); Verhaegen and Westwick (1996) and Al-Duwaish and Karim (1997).

The method proposed in this Chapter consists of two stages: First, it takes into account the information about the structure of the system in order to approximate its impulse

This chapter is based on:
Castro-Garcia, R., Agudelo, O. M., Suykens, J. A. K. (2017a). Impulse response constrained LS-SVM modeling for MIMO Hammerstein system identification. International Journal of Control. doi: 10.1080/00207179.2017.1373862. Available online at http://www. tandfonline.com/doi/abs/10.1080/00207179.2017.1373862.

response. Then, using this estimation, an LS-SVM model of the whole system is obtained. The main advantages of the proposed method include its versatility with respect to the class of systems that can be modeled as it is applicable to cases where the problem class of the nonlinearities is unknown due to the good generalization properties of LS-SVM. Another advantage is that the proposed method can be naturally used for the identification of multivariate Hammerstein systems with arbitrary numbers of inputs and outputs while other works have heavier restrictions in this regard.

The proposed method is tested in three examples through several Monte Carlo simulations. It is shown how the measurement noise (white Gaussian noise with zero mean) affects its behavior and also how its accuracy compares with other state of the art methodologies.

This chapter is organized as follows. In Section 9.2, the concepts on which the proposed system identification technique is based are explained. In Section 9.3, the method itself is presented. Section 9.4 shows the results found when applying the described methodology on three simulation examples. Finally, in Section 9.5, the conclusions are presented.

9.2 Background

9.2.1 Problem statement

Following the notation used a conversion between a time signal and a vector should be transparent, e.g. a time signal $u(t)$ with samples at $t = 0, 1, \ldots, N-1$ can be represented as a vector $\boldsymbol{u} \in \mathbb{R}^N$. Similarly, a set of p signals $u_i(t)$ with $i = 1, \ldots, p$ and samples at $t = 0, 1, \ldots, N-1$ could be represented as a matrix $\boldsymbol{U} \in \mathbb{R}^{N \times p}$. Throughout this chapter, the time and vector notations will be used interchangeably.

Given the structure of Hammerstein systems, the system to be identified is of the form

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{f}(\boldsymbol{U}), \tag{9.1}$$

where $\boldsymbol{H}$ is an impulse response matrix representing the linear part and $\boldsymbol{f}(\cdot)$ is the nonlinear part.

The impulse response matrix of a SISO linear time invariant (LTI) system can be constructed as follows (Ljung, 1999):

$$\boldsymbol{H} = \begin{bmatrix} y_{imp}(0) & 0 & 0 & \cdots & 0 \\ y_{imp}(1) & y_{imp}(0) & 0 & \cdots & 0 \\ y_{imp}(2) & y_{imp}(1) & y_{imp}(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{imp}(N-1) & y_{imp}(N-2) & y_{imp}(N-3) & \cdots & y_{imp}(0) \end{bmatrix}. \tag{9.2}$$

Where the vector $\boldsymbol{y}_{imp} = [y_{imp}(0), y_{imp}(1), \ldots, y_{imp}(N-1)]$ is the response of the system to an impulse input.

For a system with p inputs a matrix of the form $\boldsymbol{U} \in \mathbb{R}^{N\times p}$, where each column represents an input signal, can be used. It is assumed that there will be as many intermediate variables as inputs, therefore the intermediate variables will be $\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{U})$ with $\boldsymbol{f}_i : \mathbb{R}^{N\times p} \to \mathbb{R}^N$ for $i = 1, \ldots, p$. Note then that for such a system $\boldsymbol{f} : \mathbb{R}^{N\times p} \to \mathbb{R}^{pN}$ with $\boldsymbol{f}(\boldsymbol{U}) = [\boldsymbol{f}_1(\boldsymbol{U})^\top, \ldots, \boldsymbol{f}_p(\boldsymbol{U})^\top]^\top$. For a system with r outputs, $\boldsymbol{y} \in \mathbb{R}^{rN}$. Finally, for a system with p inputs and r outputs the impulse response matrix $\boldsymbol{H} \in \mathbb{R}^{rN\times pN}$ is as follows:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} & \cdots & \boldsymbol{H}_{1p} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} & \cdots & \boldsymbol{H}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_{r1} & \boldsymbol{H}_{r2} & \cdots & \boldsymbol{H}_{rp} \end{bmatrix}, \tag{9.3}$$

with each $\boldsymbol{H}_{ij}$ as defined in (9.2) corresponding to the impulse response matrix from the jth input to the ith output. It is assumed that $\boldsymbol{f}_i(\boldsymbol{0}_{N\times p}) = \boldsymbol{0}_N$.

9.2.2 Impulse response of MIMO Hammerstein Systems

In Jeng and Huang (2008) a method for estimating the impulse response of a MIMO Hammerstein is introduced. Such method, where specially designed two level signals are used to take advantage of the inherent structure of a Hammerstein system, is adapted in this chapter.

In order to excite the system for the identification of the LTI subsystem the inputs are divided in as many stages as inputs. At each stage a different input is a Pseudo Random Binary Signal (PRBS) switching between zero and a non-zero constant while all the other inputs are kept at zero. For a 2-input system, an illustration example is provided in Fig. 9.1. Given these inputs, the intermediate variables will be synchronized with the changing input at each stage and will have values of either 0 or a nonzero constant. This means that the intermediate variables will also be PRBS.

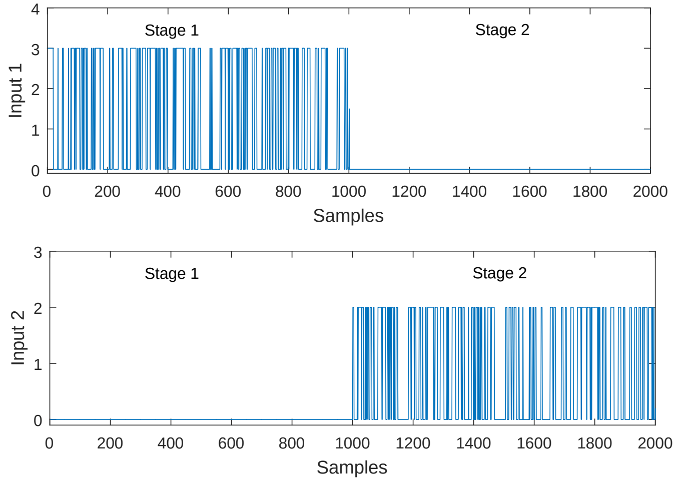


Figure 9.1: Example of the inputs for the identification of the LTI subsystem of a system with 2 inputs.

Without loss of generality, consider a system with 2 inputs $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ as the one represented in Fig. 9.2. For the identification of the linear part, in this system there will be 2 stages then. In the first stage $u_{1i} \in \{0, \bar{u}_1\}$ and $u_{2i} = 0$ with $i = 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$ and $\bar{u}_1$ the nonzero constant of the PRBS part of $\boldsymbol{u}_1$. Similarly, for the second stage $u_{1i} = 0$ and $u_{2i} \in \{0, \bar{u}_2\}$ for $i = \left\lfloor \frac{N}{2} \right\rfloor + 1, \ldots, 2 \left\lfloor \frac{N}{2} \right\rfloor$ and $\bar{u}_2$ the nonzero constant of the PRBS part of $\boldsymbol{u}_2$. For such an excitation, the corresponding intermediate variables will be as stated in (9.4) for the first stage and as stated in (9.5) for the second one:

$$\begin{array}{l} x_{1i} \in \{0, f_1(\bar{u}_1, 0)\} \\ x_{2i} \in \{0, f_2(\bar{u}_1, 0)\} \end{array}, i = 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor, \tag{9.4}$$

$$\begin{array}{l} x_{1i} \in \{0, f_1(0, \bar{u}_2)\} \\ x_{2i} \in \{0, f_2(0, \bar{u}_2)\} \end{array}, i = \left\lfloor \frac{N}{2} \right\rfloor + 1, \ldots, 2 \left\lfloor \frac{N}{2} \right\rfloor. \tag{9.5}$$

Let us define now $\boldsymbol{H}_{ij} \in \mathbb{R}^{N \times N}$ with $i = 1, \ldots, r$ and $j = 1, \ldots, p$ as the impulse response matrices corresponding to the different LTI blocks conforming the linear part of a system with p inputs and r outputs. Note that for $p = 2$ and $r = 2$ the system can

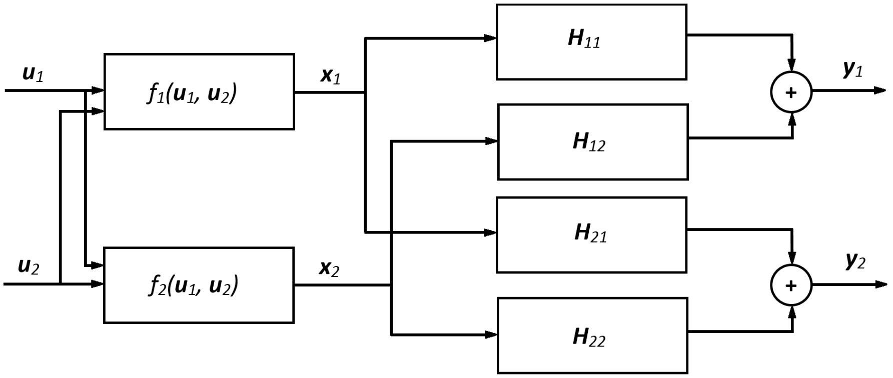


Figure 9.2: A MIMO Hammerstein system with two inputs and two outputs. $\boldsymbol{H}_{ij}$ are impulse response matrices corresponding to the linear dynamical systems of the i^{th} output and the j^{th} intermediate variable. $\boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $\boldsymbol{f}_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$ are static nonlinearities.

then be represented as in (9.6) provided that both a_1 and a_2 are different from 0.

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) \\ \boldsymbol{f}_2(\boldsymbol{u}_1, \boldsymbol{u}_2) \end{bmatrix}, \\ &= \begin{bmatrix} \frac{\boldsymbol{H}_{11}}{a_1} & \frac{\boldsymbol{H}_{12}}{a_2} \\ \frac{\boldsymbol{H}_{21}}{a_1} & \frac{\boldsymbol{H}_{22}}{a_2} \end{bmatrix} \begin{bmatrix} a_1\boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) \\ a_2\boldsymbol{f}_2(\boldsymbol{u}_1, \boldsymbol{u}_2) \end{bmatrix} \end{aligned} \tag{9.6}$$

with $\boldsymbol{f}_1 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $\boldsymbol{f}_2 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ the static nonlinear mappings; $\boldsymbol{y}_1 \in \mathbb{R}^N$ and $\boldsymbol{y}_2 \in \mathbb{R}^N$ the outputs and $\boldsymbol{u}_1,\ \boldsymbol{u}_2 \in \mathbb{R}^N$ the inputs of the system.

Equation (9.6) clearly shows that there could be a rescaling in either the linear or nonlinear parts and the other would have to compensate to keep the input-output relation. This is a known fact in the identification of this kind of systems (Boyd & Chua, 1983), where a rescaling of the blocks has no effect on the input-output behavior of the Hammerstein system (i.e. any pair of $\{\boldsymbol{f}(\boldsymbol{U})/\eta,\ \eta\boldsymbol{H}\}$ with $\eta \neq 0$ would yield identical input and output measurements).

Let us now define $a_1 = \bar{u}_1/f_1(\bar{u}_1, 0)$ and $a_2 = \bar{u}_2/f_2(0, \bar{u}_2)$. This definition necessarily means that during the first stage $a_1\boldsymbol{x}_1 = \boldsymbol{u}_1$ and $a_2\boldsymbol{x}_2 = \rho_2\boldsymbol{u}_1$ with $\rho_2 = a_2 f_2(\bar{u}_1, 0)/\bar{u}_1$. In a similar way, for the second stage $a_1\boldsymbol{x}_1 = \rho_1\boldsymbol{u}_2$ and $a_2\boldsymbol{x}_2 = \boldsymbol{u}_2$ with $\rho_1 = a_1 f_1(0, \bar{u}_2)/\bar{u}_2$.

From (9.6) then:

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{H}_{11}\boldsymbol{x}_1 + \boldsymbol{H}_{12}\boldsymbol{x}_2, \\ \boldsymbol{y}_2 &= \boldsymbol{H}_{21}\boldsymbol{x}_1 + \boldsymbol{H}_{22}\boldsymbol{x}_2 \end{aligned} \tag{9.7}$$

for the first stage this means

$$\begin{aligned} \boldsymbol{y}_1 &= \left[\frac{\boldsymbol{H}_{11}}{a_1} + \rho_2\frac{\boldsymbol{H}_{12}}{a_2}\right]\boldsymbol{u}_1 = \boldsymbol{H}_1^{(1)}\boldsymbol{u}_1, \\ \boldsymbol{y}_2 &= \left[\frac{\boldsymbol{H}_{21}}{a_1} + \rho_2\frac{\boldsymbol{H}_{22}}{a_2}\right]\boldsymbol{u}_1 = \boldsymbol{H}_2^{(1)}\boldsymbol{u}_1. \end{aligned} \tag{9.8}$$

Similarly, for the second stage

$$\begin{aligned} \boldsymbol{y}_1 &= \left[\rho_1\frac{\boldsymbol{H}_{11}}{a_1} + \frac{\boldsymbol{H}_{12}}{a_2}\right]\boldsymbol{u}_{\mathbf{2}} = \boldsymbol{H}_1^{(2)}\boldsymbol{u}_2, \\ \boldsymbol{y}_2 &= \left[\rho_1\frac{\boldsymbol{H}_{21}}{a_1} + \frac{\boldsymbol{H}_{22}}{a_2}\right]\boldsymbol{u}_2 = \boldsymbol{H}_2^{(2)}\boldsymbol{u}_{\mathbf{2}}. \end{aligned} \tag{9.9}$$

Let us define now

$$\begin{aligned} \boldsymbol{Q} &= \begin{bmatrix} \boldsymbol{H}_1^{(1)} & \boldsymbol{H}_1^{(2)} \\ \boldsymbol{H}_2^{(1)} & \boldsymbol{H}_2^{(2)} \end{bmatrix}, \\ \boldsymbol{H} &= \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} \end{bmatrix}, \\ \boldsymbol{P} &= \begin{bmatrix} \frac{\boldsymbol{I}}{a_1} & \frac{\rho_1\boldsymbol{I}}{a_1} \\ \frac{\rho_2\boldsymbol{I}}{a_2} & \frac{\boldsymbol{I}}{a_2} \end{bmatrix}. \end{aligned} \tag{9.10}$$

Therefore

$$\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{P}. \tag{9.11}$$

From (9.8) and (9.9) it is clear that $\boldsymbol{H}_1^{(1)}$, $\boldsymbol{H}_2^{(1)}$, $\boldsymbol{H}_1^{(2)}$ and $\boldsymbol{H}_2^{(2)}$ can be directly calculated as $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are known. In this chapter the least squares method was used to calculate $\boldsymbol{Q}$.

In order to calculate the impulse responses corresponding to the LTI blocks (i.e. $\boldsymbol{H}$), the only missing element now is $\boldsymbol{P}$.

Theorem 2. *Let us define the following matrix*

$$\tilde{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{I} & \tilde{\rho}_1\boldsymbol{I} \\ \tilde{\rho}_2\boldsymbol{I} & \boldsymbol{I} \end{bmatrix},$$

with

$$\begin{aligned} \tilde{\rho}_1 &= \beta_1\rho_1, \\ \tilde{\rho}_2 &= \beta_2\rho_2. \end{aligned}$$

Here $\beta_1, \beta_2 \in \mathbb{R}$ with $\beta_1 \neq 0$ and $\beta_2 \neq 0$. β_1 is the proportion between the actual ρ_1, which is unknown, and the chosen $\tilde{\rho}_1$. Similarly, β_2 is the proportion between the actual ρ_2 and the chosen $\tilde{\rho}_2$.

Provided that $\tilde{\rho}_1\tilde{\rho}_2 \neq 1$, matrix $\tilde{\boldsymbol{P}}$ can be used in place of $\boldsymbol{P}$ in (9.11). *From an input-output perspective the response of the identified MIMO Hammerstein system is then the same.*

Proof.

$$\tilde{\boldsymbol{P}}^{-1} = \frac{1}{1-\beta_1\beta_2\rho_1\rho_2}\begin{bmatrix} \boldsymbol{I} & -\beta_1\rho_1\boldsymbol{I} \\ -\beta_2\rho_2\boldsymbol{I} & \boldsymbol{I} \end{bmatrix}.$$

As $\boldsymbol{Q}$ is known and instead of the actual $\boldsymbol{P}$ only $\tilde{\boldsymbol{P}}$ is available, from (9.11) an approximation to the linear block $\hat{\boldsymbol{H}}$ is found:

$$\begin{aligned} \hat{\boldsymbol{H}} &= \boldsymbol{Q}\tilde{\boldsymbol{P}}^{-1}, \\ &= \boldsymbol{H}\boldsymbol{P}\tilde{\boldsymbol{P}}^{-1}, \\ &= \boldsymbol{H}\begin{bmatrix} k_{11}\boldsymbol{I} & k_{12}\boldsymbol{I} \\ k_{21}\boldsymbol{I} & k_{22}\boldsymbol{I} \end{bmatrix}, \\ &= \boldsymbol{H}\boldsymbol{K}, \end{aligned}$$

with

$$k_{11} = \frac{1-\beta_2\rho_1\rho_2}{a_1(1-\beta_1\beta_2\rho_1\rho_2)},\ k_{12} = \frac{(1-\beta_1)\rho_1}{a_1(1-\beta_1\beta_2\rho_1\rho_2)},$$

$$k_{21} = \frac{(1-\beta_2)\rho_2}{a_2(1-\beta_1\beta_2\rho_1\rho_2)},\ k_{22} = \frac{1-\beta_1\rho_1\rho_2}{a_2(1-\beta_1\beta_2\rho_1\rho_2)}.$$

Note that in the case that $\rho_1 = \tilde{\rho}_1$ and $\rho_2 = \tilde{\rho}_2$, then $\beta_1 = \beta_2 = 1$ and $\boldsymbol{K} = \begin{bmatrix} \frac{\boldsymbol{I}}{a_1} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\boldsymbol{I}}{a_2} \end{bmatrix}$.

If $\boldsymbol{H}\boldsymbol{K}$ is used as an approximation to the linear part, all that is required for modeling the system, from an input-output perspective, is that the intermediate variables $\boldsymbol{x}$ are modified by $\boldsymbol{K}^{-1}$, this is:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{K}\boldsymbol{K}^{-1}\boldsymbol{x}.$$

The only remaining consideration then is whether $\boldsymbol{K}$ is actually invertible. For $\boldsymbol{K}$ not invertible:

$$\begin{array}{rrcl}
 & k_{21}k_{12} & = & k_{11}k_{22} \\
\Rightarrow & \frac{1-\beta_2\rho_1\rho_2}{a_1(1-\beta_1\beta_2\rho_1\rho_2)}\frac{1-\beta_1\rho_1\rho_2}{a_2(1-\beta_1\beta_2\rho_1\rho_2)} & = & \frac{(1-\beta_1)\rho_1}{a_1(1-\beta_1\beta_2\rho_1\rho_2)}\frac{(1-\beta_2)\rho_2}{a_2(1-\beta_1\beta_2\rho_1\rho_2)} \\
\Rightarrow & 1-\beta_2\rho_1\rho_2-\beta_1\rho_1\rho_2+\beta_1\beta_2(\rho_1\rho_2)^2 & = & (1-\beta_1-\beta_2+\beta_1\beta_2)\rho_2\rho_1 \\
\Rightarrow & 1+\beta_1\beta_2(\rho_1\rho_2)^2 & = & \rho_1\rho_2+\beta_1\beta_2\rho_1\rho_2 \\
\Rightarrow & \frac{1}{\rho_1\rho_2}+\beta_1\beta_2\rho_1\rho_2 & = & 1+\beta_1\beta_2 \\
\Rightarrow & \frac{1}{\rho_1\rho_2}-1 & = & \beta_1\beta_2(1-\rho_1\rho_2) \\
\Rightarrow & \frac{1-\rho_1\rho_2}{\rho_1\rho_2(1-\rho_1\rho_2)} & = & \beta_1\beta_2 \\
\Rightarrow & \frac{1}{\rho_1\rho_2} & = & \beta_1\beta_2 \\
\Rightarrow & 1 & = & \tilde{\rho}_1\tilde{\rho}_2.
\end{array}$$

It can be seen that the non-invertible case of $\boldsymbol{K}$ is always avoided as $\tilde{\rho}_1\tilde{\rho}_2 \neq 1$ is already a constraint for selecting $\tilde{\rho}_1$ and $\tilde{\rho}_2$. □

The number of inputs determines the number of existing ρ (and $\tilde{\rho}$) variables in the system. For a system with p inputs this number is $p(p-1)$. In Theorem 2 it was shown, for a system with 2 inputs and 2 outputs, that as long as $\tilde{\rho}_1\tilde{\rho}_2 \neq 1$ the chosen values of $\tilde{\rho}_1$ and $\tilde{\rho}_2$ will not affect the response of the identified MIMO Hammerstein system from an input-output perspective. In order to guarantee a proper selection of the $\tilde{\rho}$ values in systems with more inputs, an equivalent restriction must be made, this is: The chosen $\tilde{\rho}_i$ for $i = 1, \ldots, p(p-1)$ must be such that $\det(\tilde{\boldsymbol{P}}) \neq 0$.

In Fig. 9.3 a graphical representation of the model to be found is presented.

9.3 Proposed Method

9.3.1 MIMO case

For illustrative purposes and without loss of generality we will show the case where the system has two inputs $\boldsymbol{u}_1,\ \boldsymbol{u}_2 \in \mathbb{R}^N$, two intermediate variables $\boldsymbol{x}_1,\ \boldsymbol{x}_2 \in \mathbb{R}^N$ and two outputs $\boldsymbol{y}_1,\ \boldsymbol{y}_2 \in \mathbb{R}^N$ (see Fig. 9.2). In Section 9.3.2 this will be extended to a more general case with p inputs and r outputs.

In this system, the nonlinear part will be approximated by two separate expressions: $\hat{\boldsymbol{x}}_1 = \hat{\boldsymbol{f}}_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $\hat{\boldsymbol{x}}_2 = \hat{\boldsymbol{f}}_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$. Also the matrix $\hat{\boldsymbol{H}}$ (i.e. see Theorem 2) will approximate $G(q)$ in the time domain and is composed by the approximation of the impulse response of the LTI blocks that combine the intermediate variables into the

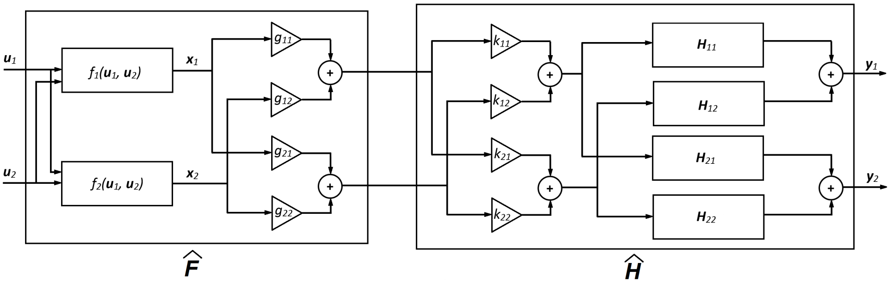


Figure 9.3: The model to be estimated. Note that once $\hat{\boldsymbol{H}}$ is calculated, $\hat{\boldsymbol{F}}$ can compensate the difference between $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$. Here $g_{11} = \frac{k_{22}}{k_{11}k_{22}-k_{12}k_{21}}$, $g_{12} = -\frac{k_{12}}{k_{11}k_{22}-k_{12}k_{21}}$, $g_{21} = -\frac{k_{21}}{k_{11}k_{22}-k_{12}k_{21}}$ and $g_{22} = \frac{k_{11}}{k_{11}k_{22}-k_{12}k_{21}}$.

outputs.

$$\hat{\boldsymbol{H}} = \begin{bmatrix} \hat{\boldsymbol{H}}_{11} & \hat{\boldsymbol{H}}_{12} \\ \hat{\boldsymbol{H}}_{21} & \hat{\boldsymbol{H}}_{22} \end{bmatrix}. \tag{9.12}$$

With the elements above, we can derive an expression to incorporate the linear part into the calculation of the whole model. Let us define

$$\begin{aligned} \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} \in \mathbb{R}^{2N}, \\ \boldsymbol{M}_1 &= \begin{bmatrix} \hat{\boldsymbol{H}}_{11} \\ \hat{\boldsymbol{H}}_{21} \end{bmatrix} \in \mathbb{R}^{2N \times N}, \\ \boldsymbol{M}_2 &= \begin{bmatrix} \hat{\boldsymbol{H}}_{12} \\ \hat{\boldsymbol{H}}_{22} \end{bmatrix} \in \mathbb{R}^{2N \times N}, \\ \boldsymbol{e} &= \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} \in \mathbb{R}^{2N}, \end{aligned} \tag{9.13}$$

where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ correspond to the errors associated to the outputs $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ respectively.

Now, let the cost function be defined as

$$\begin{aligned} \min_{\boldsymbol{w}_1,\boldsymbol{w}_2,b_1,b_2,\boldsymbol{e}_1,\boldsymbol{e}_2} \quad & \boldsymbol{J} = \tfrac{1}{2}\boldsymbol{w}_1^\top \boldsymbol{w}_1 + \tfrac{1}{2}\boldsymbol{w}_2^\top \boldsymbol{w}_2 + \tfrac{\gamma_1}{2}\boldsymbol{e}_1^\top \boldsymbol{e}_1 + \tfrac{\gamma_2}{2}\boldsymbol{e}_2^\top \boldsymbol{e}_2 \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{M}_1\left(\boldsymbol{\Phi}_1^\top \boldsymbol{w}_1 + \mathbf{1}_N b_1\right) + \boldsymbol{M}_2\left(\boldsymbol{\Phi}_2^\top \boldsymbol{w}_2 + \mathbf{1}_N b_2\right) + \boldsymbol{e}. \end{aligned} \tag{9.14}$$

Here b_1 and b_2 are the bias terms for each nonlinear function (i.e. $\hat{\boldsymbol{f}}_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $\hat{\boldsymbol{f}}_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$), $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are the aggregation of columns $\varphi_1(\boldsymbol{x}_i)$ and $\varphi_2(\boldsymbol{x}_i)$ respectively (i.e. the functions mapping the inputs to a higher dimensional feature space) with $i = 1, \ldots, N$, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are the weight vectors of the nonlinear functions and $\mathbf{1}_N$ is a column vector of ones with length N (the length of the data set).

From the Lagrangian

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}_1, \boldsymbol{w}_2, b_1, b_2, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}_1^\top \boldsymbol{w}_1 + \frac{1}{2}\boldsymbol{w}_2^\top \boldsymbol{w}_2 + \frac{\gamma_1}{2}\boldsymbol{e}_1^\top \boldsymbol{e}_1 + \frac{\gamma_2}{2}\boldsymbol{e}_2^\top \boldsymbol{e}_2 \\ -\boldsymbol{\alpha}^\top \left(\boldsymbol{M}_1 \left(\boldsymbol{\Phi}_1^\top \boldsymbol{w}_1 + \mathbf{1}_N b_1\right) + \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{w}_2 + \mathbf{1}_N b_2\right) + \boldsymbol{e} - \boldsymbol{y}\right)\end{aligned} \tag{9.15}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^{2N}$ the Lagrange multipliers, the optimality conditions are derived:

$$\begin{array}{llll} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_1} & = 0 \rightarrow & \boldsymbol{w}_1 = & \boldsymbol{\Phi}_1 \boldsymbol{M}_1^\top \boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_2} & = 0 \rightarrow & \boldsymbol{w}_2 = & \boldsymbol{\Phi}_2 \boldsymbol{M}_2^\top \boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial b_1} & = 0 \rightarrow & 0 = & \mathbf{1}_N^\top \boldsymbol{M}_1^\top \boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial b_2} & = 0 \rightarrow & 0 = & \mathbf{1}_N^\top \boldsymbol{M}_2^\top \boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{e}} & = 0 \rightarrow & \boldsymbol{\alpha} = & \boldsymbol{\Gamma} \boldsymbol{e} \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} & = 0 \rightarrow & \boldsymbol{y} = & \boldsymbol{M}_1 \left(\boldsymbol{\Phi}_1^\top \boldsymbol{w}_1 + \mathbf{1}_N b_1\right) + \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{w}_2 + \mathbf{1}_N b_2\right) + \boldsymbol{e}, \end{array} \tag{9.16}$$

with $\boldsymbol{\Gamma} = \text{diag}\left(\left[\gamma_1 \mathbf{1}_N^\top, \gamma_2 \mathbf{1}_N^\top\right]^\top\right)$.

The last equation in (9.16) can be rewritten by replacing $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and $\boldsymbol{e}$ as

$$\boldsymbol{y} = \boldsymbol{M}_1 \left(\boldsymbol{\Phi}_1^\top \boldsymbol{\Phi}_1 \boldsymbol{M}_1^\top \boldsymbol{\alpha} + \mathbf{1}_N b_1\right) + \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{\Phi}_2 \boldsymbol{M}_2^\top \boldsymbol{\alpha} + \mathbf{1}_N b_2\right) + \boldsymbol{\Gamma}^{-1} \boldsymbol{\alpha}. \tag{9.17}$$

Note again that Mercer's theorem can be used and therefore kernel functions $\boldsymbol{\Omega}_{ij}^{(1)} = k_1(\boldsymbol{u}_{train,i}, \boldsymbol{u}_{train,j})$ and $\boldsymbol{\Omega}_{ij}^{(2)} = k_2(\boldsymbol{u}_{train,i}, \boldsymbol{u}_{train,j})$ represent the kernel matrices $\boldsymbol{\Omega}^{(1)}$ and $\boldsymbol{\Omega}^{(2)}$ correspondingly. Note that this implies that $\boldsymbol{\Omega}^{(1)} = \boldsymbol{\Phi}_1^\top \boldsymbol{\Phi}_1$ and $\boldsymbol{\Omega}^{(2)} = \boldsymbol{\Phi}_2^\top \boldsymbol{\Phi}_2$.

The following linear system is finally obtained

$$\begin{bmatrix} 0 & 0 & \mathbf{1}_N^\top \boldsymbol{M}_1^\top \\ 0 & 0 & \mathbf{1}_N^\top \boldsymbol{M}_2^\top \\ \boldsymbol{M}_1 \mathbf{1}_N & \boldsymbol{M}_2 \mathbf{1}_N & \boldsymbol{M}_1 \boldsymbol{\Omega}^{(1)} \boldsymbol{M}_1^\top + \boldsymbol{M}_2 \boldsymbol{\Omega}^{(2)} \boldsymbol{M}_2^\top + \boldsymbol{\Gamma}^{-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{y} \end{bmatrix}. \tag{9.18}$$

The resulting model for a new input $\boldsymbol{U}_{new} \in \mathbb{R}^{N_n \times p}$ is then:

$$\hat{\boldsymbol{y}} = \bar{\boldsymbol{M}}_1 \mathbf{1}_N b_1 + \bar{\boldsymbol{M}}_2 \mathbf{1}_N b_2 + \left(\bar{\boldsymbol{M}}_1 \boldsymbol{K}^{(1)} \breve{\boldsymbol{M}}_1^\top + \bar{\boldsymbol{M}}_2 \boldsymbol{K}^{(2)} \breve{\boldsymbol{M}}_2^\top\right) \boldsymbol{\alpha}, \tag{9.19}$$

with $\boldsymbol{U}_{train}$ the set of inputs used to train the model, $\boldsymbol{K}_{ij}^{(1)} = k_1(\boldsymbol{u}_{train,i}, \boldsymbol{u}_{new,j})$ and $\boldsymbol{K}_{ij}^{(2)} = k_2(\boldsymbol{u}_{train,i}, \boldsymbol{u}_{new,j})$ with $i = 1, \ldots, N$, $j = 1, \ldots, N_n$ and therefore

$\boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)} \in \mathbb{R}^{N\times N_n}$. $\bar{\boldsymbol{M}}_1, \bar{\boldsymbol{M}}_2 \in \mathbb{R}^{2N_n\times N}$ and $\breve{\boldsymbol{M}}_1, \breve{\boldsymbol{M}}_2 \in \mathbb{R}^{2N\times N_n}$. Note that if $N \neq N_n$ then $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ have to be truncated or expanded to generate $\bar{\boldsymbol{M}}_1, \bar{\boldsymbol{M}}_2, \breve{\boldsymbol{M}}_1, \breve{\boldsymbol{M}}_2$ and comply with the new dimensions. Given the way these matrices are constructed, this should be straightforward.

9.3.2 General case

The method's formulation can be extended to a p inputs and r outputs case easily though the solution of such systems becomes computationally expensive very quickly as the number of inputs and/or outputs increases. For this extension, the cost function in (9.14) is rewritten leading to a rewritting of (9.18).

$$\begin{aligned} \min_{\boldsymbol{w}_i, b_i, \boldsymbol{e}_j} \quad & \boldsymbol{J} = \sum_{i=1}^{p} \frac{1}{2} \boldsymbol{w}_i^\top \boldsymbol{w}_i + \sum_{j=1}^{r} \frac{\gamma_j}{2} \boldsymbol{e}_j^\top \boldsymbol{e}_j \\ \text{subject to} \quad & \boldsymbol{y} = \sum_{i=1}^{p} \boldsymbol{M}_i \left(\boldsymbol{\Phi}_i^\top \boldsymbol{w}_i + \mathbf{1}_N b_i \right) + \boldsymbol{e}. \end{aligned} \tag{9.20}$$

Here

$$\begin{aligned} \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_r \end{bmatrix}, \text{with } \boldsymbol{y}_i \in \mathbb{R}^N \text{ and } i = 1, \ldots, r, \\ \boldsymbol{e} &= \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_r \end{bmatrix}, \text{with } \boldsymbol{e}_i \in \mathbb{R}^N \text{ and } i = 1, \ldots, r. \end{aligned} \tag{9.21}$$

Also let us define $\boldsymbol{M}_i \in \mathbb{R}^{rN\times N}$ for $i = 1, 2, \ldots, p$

$$\boldsymbol{M}_i = \begin{bmatrix} \hat{\boldsymbol{H}}_{1i} \\ \hat{\boldsymbol{H}}_{2i} \\ \vdots \\ \hat{\boldsymbol{H}}_{ri} \end{bmatrix}. \tag{9.22}$$

A matrix $\boldsymbol{M} \in \mathbb{R}^{rN\times p}$ can then be defined such that

$$\boldsymbol{M} = \left[\boldsymbol{M}_1 \mathbf{1}_N, \boldsymbol{M}_2 \mathbf{1}_N, \ldots, \boldsymbol{M}_p \mathbf{1}_N \right], \tag{9.23}$$

and a diagonal matrix $\mathbf{\Gamma} \in \mathbb{R}^{rN \times rN}$

$$\mathbf{\Gamma} \;=\; \operatorname{diag}\left(\left[\gamma_1 \mathbf{1}_N^\top, \gamma_2 \mathbf{1}_N^\top, \ldots, \gamma_r \mathbf{1}_N^\top\right]^\top\right). \tag{9.24}$$

Finally $\boldsymbol{b} \in \mathbb{R}^p$ is defined as

$$\boldsymbol{b} \;=\; \left[b_1, b_2, \ldots, b_p\right]^\top. \tag{9.25}$$

The corresponding linear system is then

$$\begin{bmatrix} \mathbf{0}_{p \times p} & \boldsymbol{M}^\top \\ \boldsymbol{M} & \mathbf{\Gamma}^{-1} + \sum_{i=1}^{p} \boldsymbol{M}_i \mathbf{\Omega}^{(i)} \boldsymbol{M}_i^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_p \\ \boldsymbol{y} \end{bmatrix}. \tag{9.26}$$

Note that for this system, $p + r$ parameters must be tuned if the RBF Gaussian kernel is used (i.e. $\sigma_1, \sigma_2, \ldots, \sigma_p$ and $\gamma_1, \gamma_2, \ldots, \gamma_r$). Also note that $\boldsymbol{\alpha}, \boldsymbol{y} \in \mathbb{R}^{rN}$.

The resulting model for a new input $\boldsymbol{U}_{new} \in \mathbb{R}^{N_n \times p}$ is then:

$$\hat{\boldsymbol{y}} = \bar{\boldsymbol{M}} \boldsymbol{b} + \left(\sum_{i=1}^{p} \bar{\boldsymbol{M}}_i \boldsymbol{K}^{(i)} \breve{\boldsymbol{M}}_i^\top \right) \boldsymbol{\alpha}, \tag{9.27}$$

with $\boldsymbol{U}_{train} \in \mathbb{R}^{N \times p}$ the set of inputs used to train the model, $\boldsymbol{K}_{lj}^{(i)} = k_i(\boldsymbol{u}_{train,l}, \boldsymbol{u}_{new,j})$ with $k_i(\cdot,\cdot)$ the kernel functions with $i = 1, \ldots, p$, $l = 1, \ldots, N$ and $j = 1, \ldots, N_n$ and therefore $\boldsymbol{K}^{(i)} \in \mathbb{R}^{N \times N_n}$. $\bar{\boldsymbol{M}}_i \in \mathbb{R}^{rN_n \times N}$, $\bar{\boldsymbol{M}} = \left[\bar{\boldsymbol{M}}_1 \mathbf{1}_N, \bar{\boldsymbol{M}}_2 \mathbf{1}_N, \ldots, \bar{\boldsymbol{M}}_p \mathbf{1}_N\right]$ and $\breve{\boldsymbol{M}}_i \in \mathbb{R}^{rN \times N_n}$. Again, note that if $N \neq N_n$ then the $\boldsymbol{M}_i$ matrices have to be truncated or expanded to generate $\bar{\boldsymbol{M}}_i, \breve{\boldsymbol{M}}_i$ and comply with the new dimensions.

In Algorithm 5 a summary of the proposed methodology is presented.

9.4 Simulation Results

9.4.1 Method steps

The proposed method was applied to three examples with two inputs and two outputs, consisting of two nonlinear functions and four LTI blocks as illustrated in Fig. 9.2. Note that in order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D. The corresponding nonlinear functions of Example 1 are given in (9.28) and plotted in Fig. 9.4.

$$f_1(u_1, u_2) = \frac{u_1^3}{10} + 0.9u_2^2, \tag{9.28a}$$

Algorithm 5 Impulse Response Constrained LS-SVM for MIMO Hammerstein System Identification.

```
Input: Multi-stage pseudo-random binary input signals for estimation of the linear
    part $\boldsymbol{U}_{LP} \in \mathbb{R}^{N \times p}$ (i.e. $p$ inputs signals of length $N$) and their corresponding $r$
    outputs $\boldsymbol{y}_{LP} \in \mathbb{R}^{rN}$. Training inputs for the LS-SVM $\boldsymbol{U}_{train} \in \mathbb{R}^{N \times p}$ and their
    corresponding outputs $\boldsymbol{y}_{train} \in \mathbb{R}^{rN}$. Validation inputs $\boldsymbol{U}_{val} \in \mathbb{R}^{N \times p}$ and their
    corresponding outputs $\boldsymbol{y}_{val} \in \mathbb{R}^{rN}$. Test inputs $\boldsymbol{U}_{test} \in \mathbb{R}^{N \times p}$.
Output: Evaluation of the test output signal $\boldsymbol{y}_{test} \in \mathbb{R}^{rN}$;
1: Use $\boldsymbol{U}_{LP}$ and $\boldsymbol{y}_{LP}$ to estimate the matrix $\boldsymbol{Q}$ as shown in (9.8),(9.9) and (9.10).
2: Estimate $\hat{\boldsymbol{H}} = \boldsymbol{Q}\tilde{\boldsymbol{P}}^{-1}$ with $\tilde{\boldsymbol{P}}$ as defined in Theorem 2.
3: Create matrices $\boldsymbol{M}_i$ with $i = 1, 2, \ldots, p$ using $\hat{\boldsymbol{H}}$ as shown in (9.22).
4: Use matrices $\boldsymbol{M}_i$ to create matrix $\boldsymbol{M}$ as in (9.23).
5: Assemble a linear system like the one in (9.26) and using $\boldsymbol{U}_{train}$, $\boldsymbol{y}_{train}$, $\boldsymbol{U}_{val}$
   and $\boldsymbol{y}_{val}$ proceed to tune the parameters $\sigma_1, \sigma_2, \ldots, \sigma_p$ and $\gamma_1, \gamma_2, \ldots, \gamma_r$.
6: Obtain $\boldsymbol{\alpha}$ and $\boldsymbol{b}$ from (9.26) using the estimated parameters.
7: Apply the found model to $\boldsymbol{U}_{test}$ to obtain $\hat{\boldsymbol{y}}_{test}$ as in (9.27).
8: return $\hat{\boldsymbol{y}}_{test}$.
```

$$f_2(u_1, u_2) = u_1^2 + u_2^2. \tag{9.28b}$$

The transfer functions of Example 1 are presented in (9.29) and the magnitude of their frequency response is shown in Fig. 9.5.

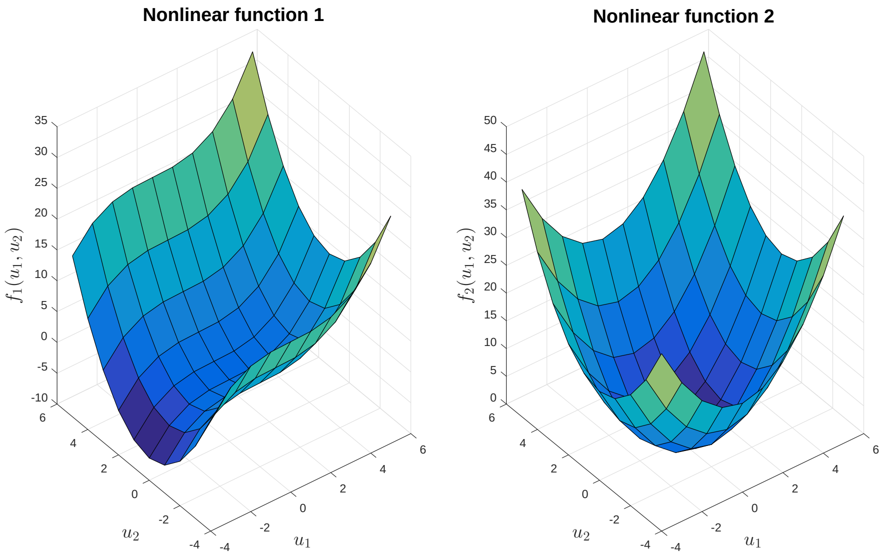
$$G_{11}(q) = \frac{0.9063}{q - 0.8187}, \tag{9.29a}$$

$$G_{12}(q) = \frac{1.572q + 1.323}{q^2 - 0.8828q + 0.6065}, \tag{9.29b}$$

$$G_{21}(q) = \frac{0.1969q^3 + 0.04616q^2 - 0.5395q - 0.1147}{q^4 - 0.9768q^3 + 0.1687q^2 - 0.268q + 0.2019}, \tag{9.29c}$$

$$G_{22}(q) = \frac{1.268q + 1.038}{q^2 - 1.452q + 0.5488}. \tag{9.29d}$$

Initially, the procedure described in Section 9.2.2 is applied in order to obtain an estimation of the impulse response of the different transfer functions. Note that as described, the parameters in matrix $\tilde{\boldsymbol{P}}$ are chosen arbitrarily and therefore the impulse responses obtained do not match exactly with the original ones. However, this will not impact the model from an input-output perspective as will be shown later. Once the impulse responses are calculated, the matrices $\boldsymbol{M}_i$ (i.e. see (9.22)) are created and the system in (9.26) can be solved. In this work, for the tuning of the parameters (i.e. σ_1,


Nonlinear function 1

Nonlinear function 2

Figure 9.4: Nonlinear functions for Example 1. (Left) $f_1(u_1, u_2)$ and (Right) $f_2(u_1, u_2)$.

σ_2, γ_1 and γ_2) Coupled Simulated Annealing (CSA) was used (Xavier-de Souza et al., 2009). In Fig. 9.6 and Fig. 9.7 the results for the estimations of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in Example 1 are shown respectively for a no-noise case and an arbitrary $\tilde{\boldsymbol{P}} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$.

In Fig. 9.8 the resulting estimations of the impulse response are shown for different values of $\tilde{\boldsymbol{P}}$ where the main diagonals are always 1. In Table 9.1 the corresponding values of $\tilde{\rho}_1$ and $\tilde{\rho}_2$ can be found together with the resulting $\%MAE$ in the test set. As can be seen, the results are quite similar despite the fact that the LTI approximations were different from the actual impulse response of the linear systems and among themselves. This was to be expected as explained in Theorem 2.

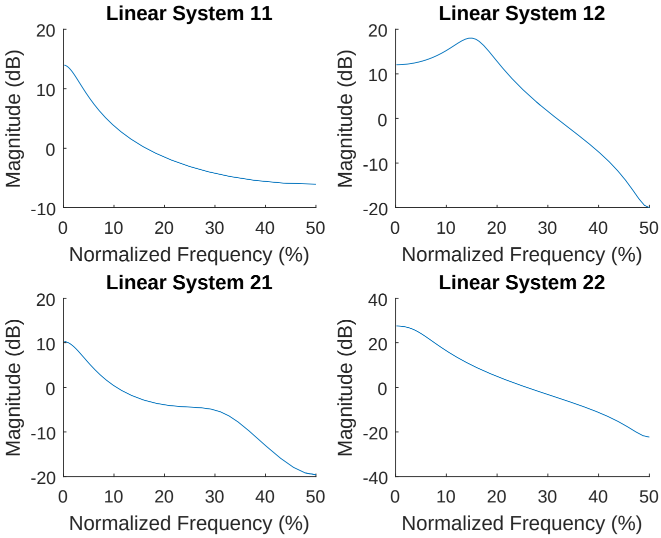


Figure 9.5: Magnitude of the frequency response of the LTI blocks for Example 1. (Upper left) $G_{11}(q)$, (Upper right) $G_{12}(q)$, (Lower left) $G_{21}(q)$ and (Lower right) $G_{22}(q)$.

Table 9.1: Different entry values for $\tilde{\rho}_1$ and $\tilde{\rho}_2$ and the resulting $\%MAE$ in the test set.

$\tilde{\rho}_1$	$\tilde{\rho}_2$	$\%MAE$ of $\boldsymbol{y}_1$	$\%MAE$ of $\boldsymbol{y}_2$
7	1	0.268060647	0.1360575
7	5	0.120215049	0.037695636
2	10	0.425032546	0.17183245
10	9	0.230242274	0.122064228
2	9	0.223370104	0.137673664
4	6	0.639722418	0.196497981
4	9	0.132787457	0.096275247
10	6	0.308935392	0.154759024
6	10	0.194847013	0.092984275
10	10	0.151705355	0.085846982

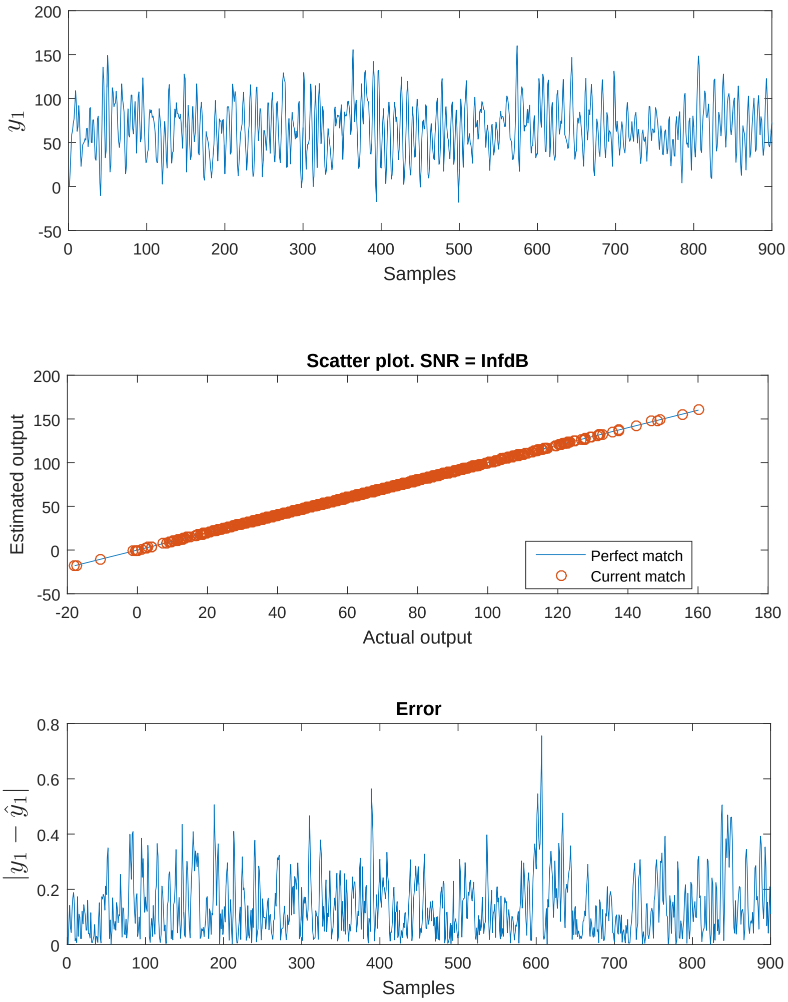


Figure 9.6: Example 1, output 1 simulation results. (Up) Real output. (Center) Scatter plot between the actual and the estimated output. (Bottom) Absolute value of the difference between the actual and the estimated output.

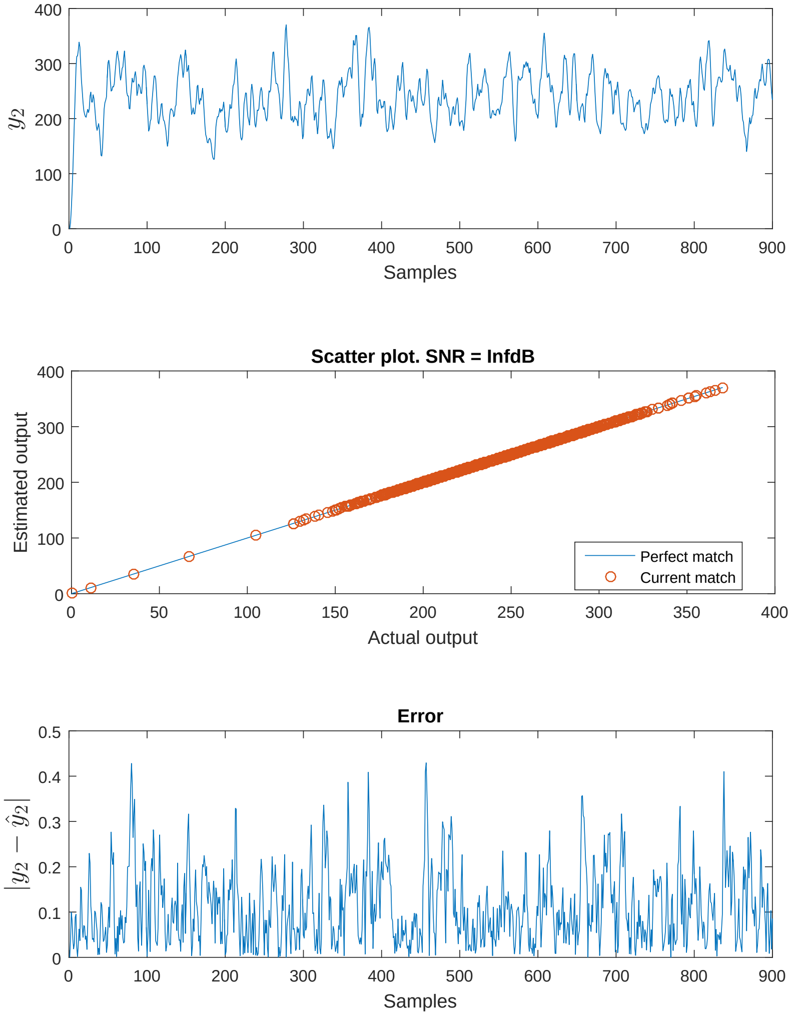


Figure 9.7: Example 1, output 2 simulation results. (Up) Real output. (Center) Scatter plot between the actual and the estimated output. (Bottom) Absolute value of the difference between the actual and the estimated output.

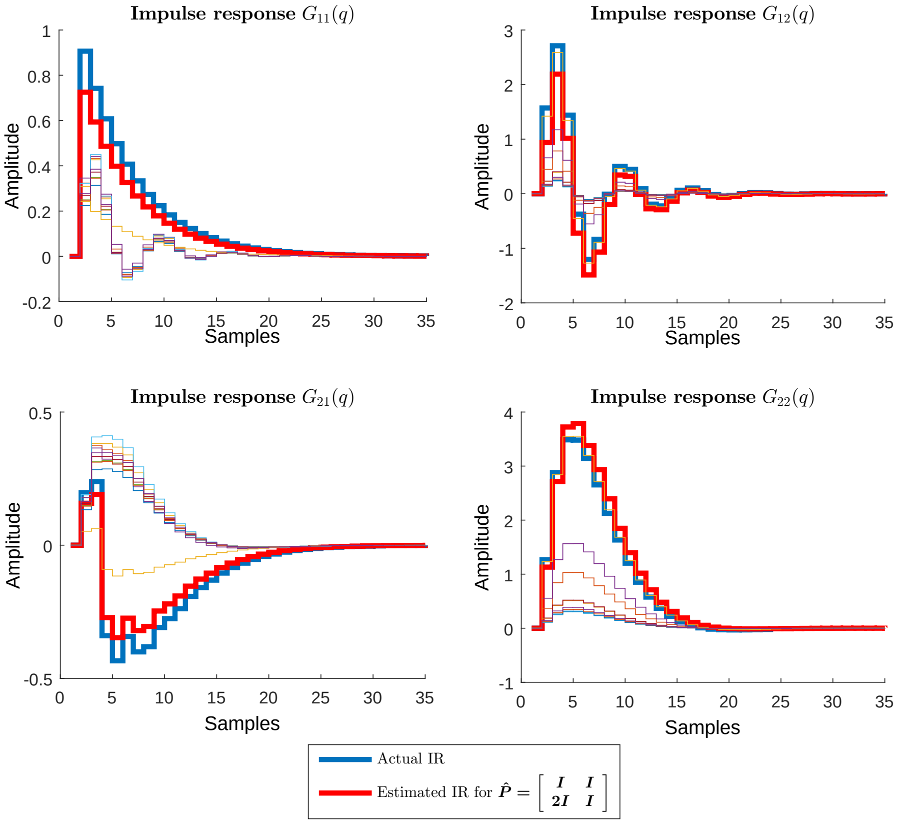


Figure 9.8: Estimation of the impulse responses. The actual impulse response appears in blue while the approximated impulse response used appears in red. The other colors represent approximations obtained through the values of $\tilde{\rho}_1$ and $\tilde{\rho}_2$ presented in Table 9.1. (Upper left) $G_{11}(q)$, (Upper right) $G_{12}(q)$, (Lower left) $G_{21}(q)$ and (Lower right) $G_{22}(q)$.

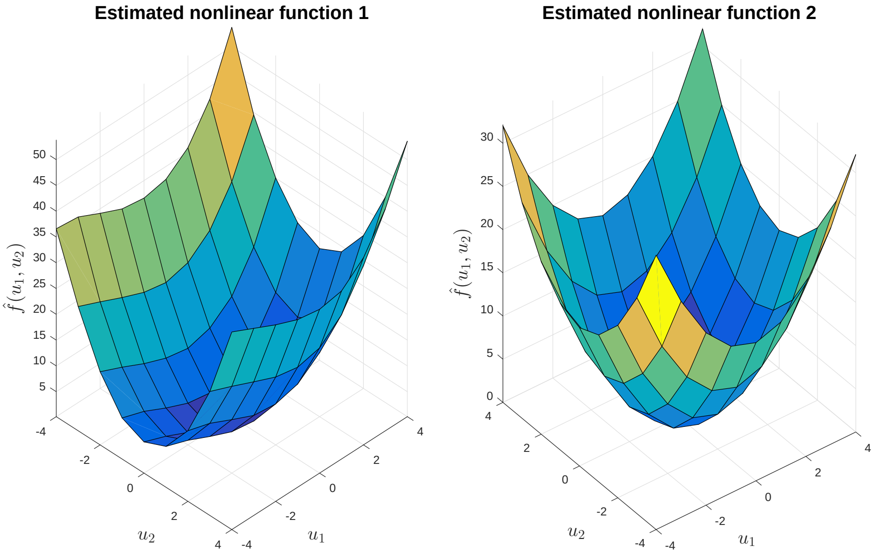


Figure 9.9: Estimated nonlinear functions for Example 1. (Left) $\hat{f}_1(u_1, u_2)$ and (Right) $\hat{f}_2(u_1, u_2)$

It is interesting to note that from (9.19) the estimated nonlinear functions can be retrieved by factorizing $\bar{\boldsymbol{M}}_1$ and $\bar{\boldsymbol{M}}_2$ such that

$$\begin{aligned} \hat{\boldsymbol{y}} &= \bar{\boldsymbol{M}}_1 \left(\mathbf{1}_N b_1 + \left(\boldsymbol{K}^{(1)} \breve{\boldsymbol{M}}_1^\top\right) \boldsymbol{\alpha}\right) + \bar{\boldsymbol{M}}_2 \left(\mathbf{1}_N b_2 + \left(\boldsymbol{K}^{(2)} \breve{\boldsymbol{M}}_2^\top\right) \boldsymbol{\alpha}\right), \\ &= \bar{\boldsymbol{M}}_1 \hat{\boldsymbol{x}}_1 + \bar{\boldsymbol{M}}_2 \hat{\boldsymbol{x}}_2, \\ &= \bar{\boldsymbol{M}}_1 \hat{\boldsymbol{f}}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) + \bar{\boldsymbol{M}}_2 \hat{\boldsymbol{f}}_2(\boldsymbol{u}_1, \boldsymbol{u}_2). \end{aligned} \tag{9.30}$$

For Example 1 the estimated nonlinear functions are presented in Fig. 9.9. Note that the estimated nonlinear functions are linear combinations of the original ones as illustrated in Fig. 9.3. When the cross terms are comparatively smaller than the corresponding diagonal terms (i.e. $g_{12} \ll g_{11}$ and $g_{22} \ll g_{21}$), the estimated functions will be more similar to the original ones.

9.4.2 Signals description

The inputs used for the proposed method had $N = 900$ points. Note that as described in Section 9.2.2 a special set of signals is used for the identification of the impulse responses of the system. In this case pseudo random binary signals ranging between 0 and 1 were used with a 5% switching probability and consisting of $2N$ points. Upper and lower limits for all inputs where set (i.e. $\boldsymbol{u}_1, \boldsymbol{u}_2 \in [-4, 5]$).

To generate the training set for the reformulated LS-SVM N is chosen such that $\sqrt{N} \in \mathbb{N}$. We select $\sqrt{N}$ values between the upper an lower limits of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ (i.e. $v_i^{(1)}$ and $v_i^{(2)}$ for $i = 1, \ldots, \sqrt{N}$). An uniform distribution covering the whole range is used (i.e. the difference between any pair of consecutive values is constant). To guarantee that all the combinations of the selected values of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are considered, the procedure is as follows:

- For each value $v_i^{(1)}$ create a vector of size $\sqrt{N}$ where all the elements are $v_i^{(1)}$.
- Randomly permute the order of the resulting vectors and concatenate them. This constitutes $\boldsymbol{u}_1 \in \mathbb{R}^N$ which is the first input of the training set.
- Randomly permute the order of the selected values $v_i^{(2)}$ to create a vector of size $\sqrt{N}$.
- Repeat the last step $\sqrt{N}$ times and concatenate the results. This constitutes $\boldsymbol{u}_2 \in \mathbb{R}^N$ which is the second input of the training set.

With this procedure it is guaranteed that all the possible combinations of $v_i^{(1)}$ and $v_i^{(2)}$ are considered, see Fig. 9.10 for an illustration.

For the tuning of the parameters described in Section 9.3.1 validation was used where the inputs where taken randomly from a uniform distribution between the upper and lower bounds defined for the training set. In this case we opted for plain validation and therefore used 10 validation sets, all of them generated in the same fashion. Note however that this could be replaced with a cross-validation scheme. In a similar way, the inputs of the test set were created by taking random values from a uniform distribution between the upper and lower bounds defined for the training set.

9.4.3 Noise effect analysis

In Section 9.4.1 the proposed method was explored step by step through Example 1 and to make the explanation more clear no noise was used. Nevertheless, for any method to be useful for system identification it is expected to be robust against noise. To show

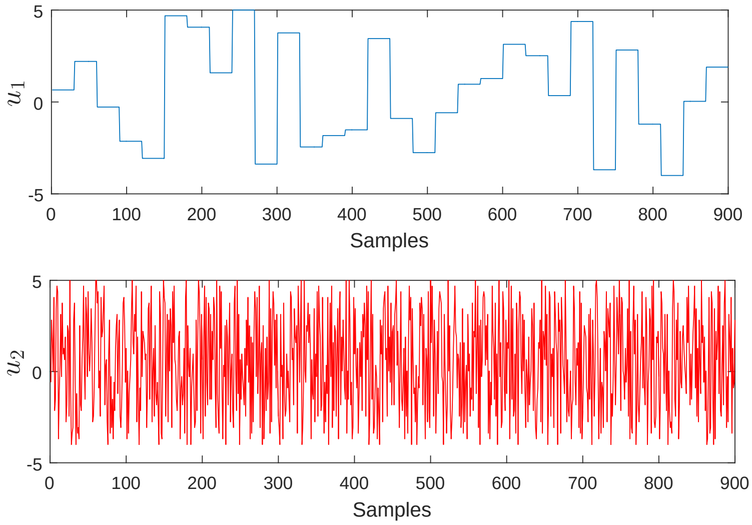


Figure 9.10: Training set for the LS-SVM part.

how the proposed method behaves in the presence of noise, Examples 2 and 3 are introduced and the results of 100 Monte Carlo simulations are offered for all examples and for different levels of white Gaussian noise with zero mean. For these simulations, signals as the ones described in Section 9.4.2 were employed.

For Example 2 the corresponding nonlinear functions are presented in (9.31) and plotted Fig. 9.11.

$$f_1(u_1,u_2) = \frac{u_1^3}{5} + \sin(u_2)u_2^2, \tag{9.31a}$$

$$f_2(u_1,u_2) = 10\sin(u_1) + u_2^2. \tag{9.31b}$$

The transfer functions of Example 2 are given in (9.32) and the magnitude of their frequency response is shown in Fig. 9.12.

$$G_{11}(q) = \frac{100q^3 + 300q^2 + 300q + 100}{q^3 - 2.458q^2 + 2.262q - 0.7654}, \tag{9.32a}$$

$$G_{12}(q) = \frac{18000q^2 - 32400q + 14400}{q^2 - 1.5q + 0.7225}, \tag{9.32b}$$

$$G_{21}(q) = \frac{10000q^4 - 1.884e04q^3 + 2.506e04q^2 - 1.884e04q + 1e04}{q^4 - 2.485q^3 + 2.528q^2 - 1.184q + 0.2245}, \tag{9.32c}$$

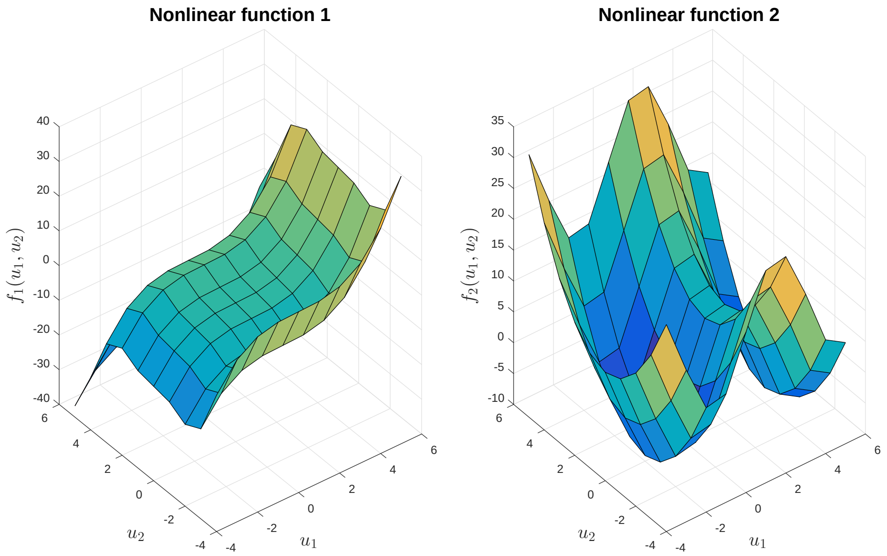


Figure 9.11: Nonlinear functions for Example 2. (Left) $f_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and (Right) $f_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$.

$$G_{22}(q) = \frac{100q^3 - 50.64q^2 - 50.64q + 100}{q^3 - 2.564q^2 + 2.218q - 0.6456}. \tag{9.32d}$$

Functions including saturation or dead-zones are generally regarded as difficult to learn. In Example 3 we include one of each of these functions. The corresponding nonlinear functions are presented in (9.33) and plotted Fig. 9.13.

$$f_1(u_1,u_2) = \begin{cases} -2 & \text{for } u_1 < -2 \\ u_1 & \text{for } -2 \leq u_1 < 2 \\ 2 & \text{for } u_1 \geq 2 \end{cases}, \tag{9.33a}$$

$$f_2(u_1,u_2) = \begin{cases} u_2 - 1.2 & \text{for } u_2 > 1.2 \\ 0 & \text{for } -1.2 \leq u_2 \leq 1.2 \\ u_2 + 1.2 & \text{for } u_2 < -1.2 \end{cases}. \tag{9.33b}$$

The transfer functions of Example 3 are given in (9.34) and the magnitude of their

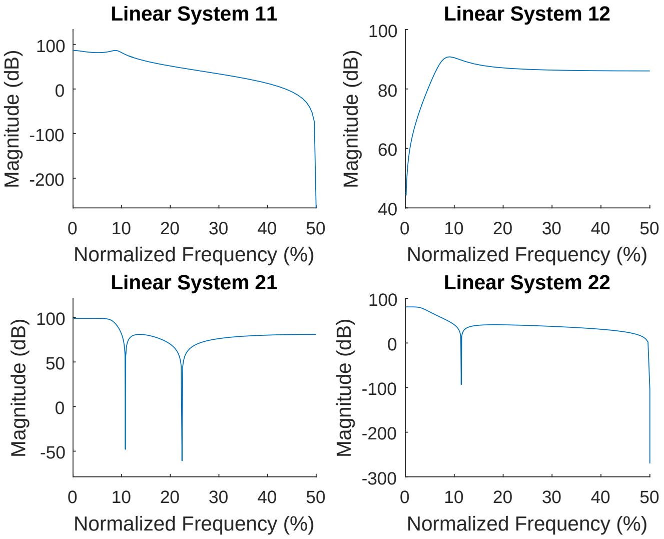


Figure 9.12: Magnitude of the frequency response of the LTI blocks for Example 2. (Upper left) $G_{11}(q)$, (Upper right) $G_{12}(q)$, (Lower left) $G_{21}(q)$ and (Lower right) $G_{22}(q)$.

frequency response is shown in Fig. 9.14.

$$G_{11}(q) = \frac{1.857}{q - 0.8607}, \tag{9.34a}$$

$$G_{12}(q) = \frac{0.001429q^2 + 0.004898q + 0.001048}{q^3 - 2.44q^2 + 1.984q - 0.5379}, \tag{9.34b}$$

$$G_{21}(q) = \frac{-0.3612q^2 + 0.1623q + 0.3408}{q^3 - 1.954q^2 + 1.627q - 0.5543}, \tag{9.34c}$$

$$G_{22}(q) = \frac{0.4599q + 0.4245}{q^2 - 1.731q + 0.7866}. \tag{9.34d}$$

The results for all examples with different levels of noise are presented in Figures 9.15 and 9.16. In Table 9.2 the resulting medians are offered to summarize the results.

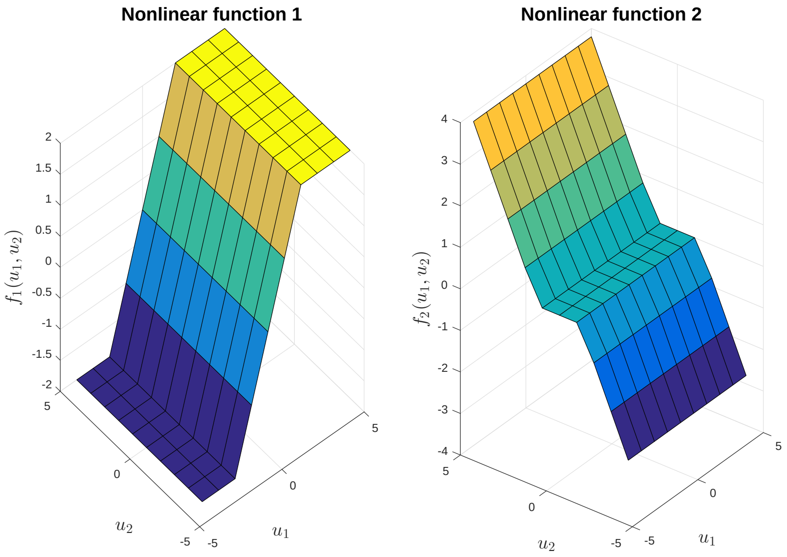


Figure 9.13: Nonlinear functions for Example 3. (Left) $f_1(u_1, u_2)$ and (Right) $f_2(u_1, u_2)$.

Table 9.2: $\%MAE$ Comparison for the proposed method. Median values are offered for 100 Monte Carlo simulations for each case.

	Example 1		Example 2		Example 3	
	$\boldsymbol{y}_1$	$\boldsymbol{y}_2$	$\boldsymbol{y}_1$	$\boldsymbol{y}_2$	$\boldsymbol{y}_1$	$\boldsymbol{y}_2$
SNR 10dB	3.8499	3.3225	4.8041	6.4876	3.7755	2.2042
SNR 20dB	1.2885	1.0098	3.235	5.7129	1.3033	1.095
SNR InfdB	0.086366	0.032617	0.028782	0.0069262	0.33918	0.77252

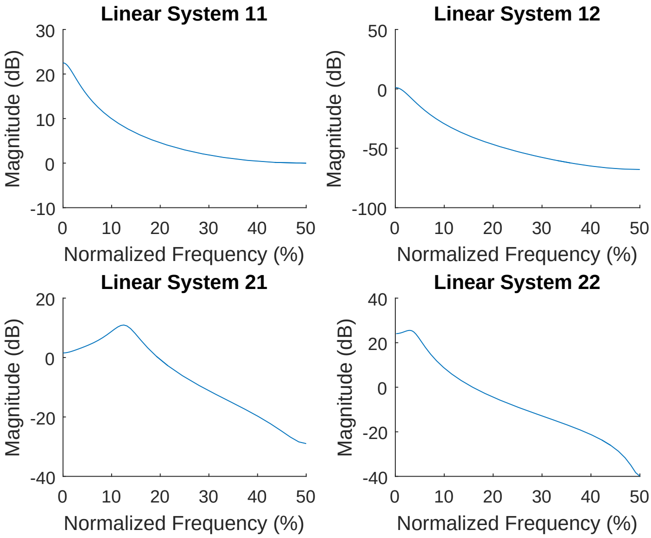


Figure 9.14: Magnitude of the frequency response of the LTI blocks for Example 3. (Upper left) $G_{11}(q)$, (Upper right) $G_{12}(q)$, (Lower left) $G_{21}(q)$ and (Lower right) $G_{22}(q)$.

9.4.4 Methods comparison

The proposed method, from now on referred to as IR-H-LS-SVM, is compared with 3 other methods for MIMO Hammerstein system identification, namely:

- NARX LS-SVM (Suykens et al., 2002).
- The method in Jeng and Huang (2008) where an approximation to the input response of the system is obtained and with it and the known output an estimation of the intermediate variable is found. Using this approximation and the known input, a mapping of the nonlinearity is done through the fitting of a polynomial. From now on, this method will be referred to as IR H-MIMO.
- Using orthonormal bases for the identification of block-oriented nonlinear systems is proposed in Gomez and Baeyens (2004). This method will be referred to as ONBF.

The NARX LS-SVM approach is not a method specifically developed for the identification of MIMO Hammerstein Systems. However, it is still relevant in this

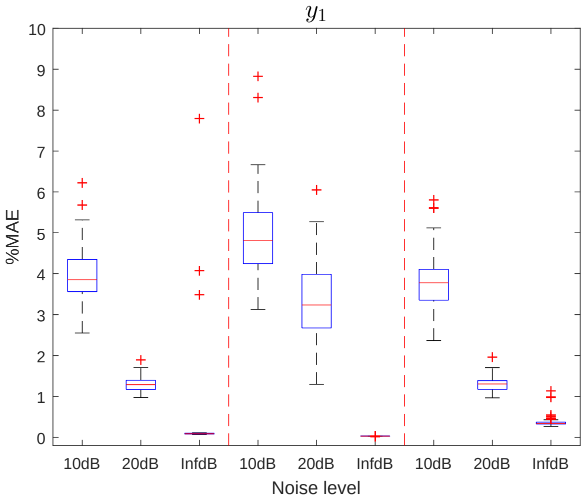


Figure 9.15: Monte Carlo simulation of the proposed method. Output 1 for Examples 1 (Left), 2 (Center) and 3 (Right).

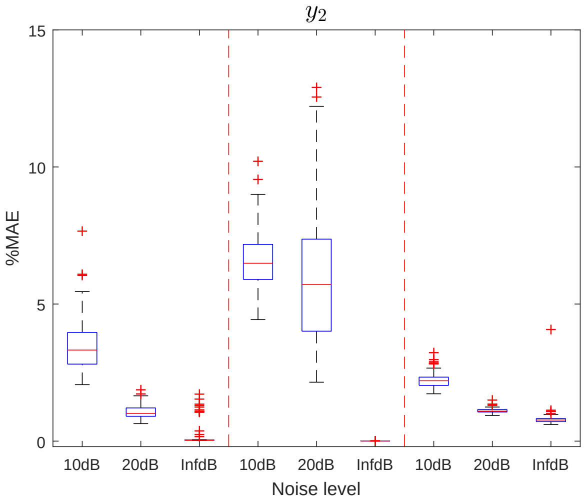


Figure 9.16: Monte Carlo simulation of the proposed method. Output 2 for Examples 1 (Left), 2 (Center) and 3 (Right).

Chapter as it allows to see how the LS-SVM framework, though powerful by itself, can be improved by including additional information from the structure of the system. The training set was generated under the same guidelines as those described in Section 9.4.2 with the exception of the number of points as in this case $N = 2500$. For the hyper parameter tuning a 10 fold cross validation scheme was used. Also, a preliminary Monte Carlo experiment was carried out to determine the optimal number of input and output lags to be used for each example. During these experiments input and output lags from 4 to 10 were tested. For Example 1, 4 lags of input and 4 lags of output were selected. For Example 2, 10 lags of input and 10 lags of output were chosen. For Example 3, 9 lags of input and 9 lags of output were used.

For the IR H-MIMO method, PRBS of 800 samples were created in order to identify the linear part. These signals followed the same guidelines described in Section 9.2.2, that is: In a first stage $\boldsymbol{u}_1$ was a PRBS and $\boldsymbol{u}_2$ was kept at 0. Then, in a second stage $\boldsymbol{u}_1$ was 0 and $\boldsymbol{u}_2$ was a PRBS. After the impulse responses were estimated, the nonlinear part was modeled. To do this, signals of 980 points were used of which the last 80 where included to make the corresponding linear system overdetermined. The initial 900 points where generated guaranteeing that all combinations of 30 points drawn from a uniform distribution between -5 and 5 were included. With these signals the nonlinearities were estimated by fitting two-dimensional polynomials with degrees 3, 7 and 9 for Examples 1, 2 and 3 respectively.

For the ONBF method, polynomial basis functions were used for identifying the nonlinearity: For Example 1 until degree 3, for Example 2 until degree 5 and for Example 3 until degree 7. Empirically, it was found that the use of simpler basis functions yielded better results for the modeling of the linear part and in consequence q^{-n} was used. For Example 1 the number of bases used was 10 and for Examples 2 and 3 it was 40. These values were set by trial and error and were the ones that offered a good trade-off between complexity and accuracy. For Example 1 1600 data points were used and for Examples 2 and 3 3600 were employed.

The results are summarized in Table 9.3. For ease of comparison, the results from Table 9.2 are included again. The best results for each case are in bold case.

As can be seen the proposed method performs quite well when compared with the other methods considered. It is the best in 12 out of the 18 cases analyzed and in general has a very good and stable performance. By this, it is meant that comparatively good results are obtained in all cases. Also, it can be seen that the proposed method is robust against the type of noise employed, this is particularly evident for the SNR $= 10dB$ case where our method performs better than the others.

Note that the class model of the nonlinear part of Example 1 is polynomial which makes its identification much easier for methods using polynomial bases (as is the case for IR H-MIMO and ONBF). However, representing the nonlinear parts of Examples 2

Table 9.3: $\%MAE$ Comparison for the different methods tested. Median values are offered for 100 Monte Carlo simulations for each case.

		Example 1		**Example 2**		**Example 3**	
		y_1	y_2	y_1	y_2	y_1	y_2
SNR 10dB	**IR H-LS-SVM**	**3.8499**	**3.3225**	4.8041	**6.4876**	**3.7755**	**2.2042**
	NARX LS-SVM	8.6475	10.5307	12.3816	9.9141	5.3544	3.2651
	IR H-MIMO	6.6443	4.9827	15.5643	20.2475	9.0841	21.1168
	ONBF	4.529	4.6303	**2.954**	6.8069	4.587	13.883
SNR 20dB	**IR H-LS-SVM**	**1.2885**	**1.0098**	3.235	5.7129	**1.3033**	**1.095**
	NARX LS-SVM	5.2555	5.5722	13.119	6.3674	3.9329	2.319
	IR H-MIMO	2.2933	2.9713	3.1664	7.2392	3.2163	6.6558
	ONBF	2.8237	1.6546	**1.5856**	**4.3417**	4.0638	6.0554
SNR InfdB	**IR H-LS-SVM**	0.086366	0.032617	**0.028782**	**0.0069262**	**0.33918**	0.77252
	NARX LS-SVM	3.9083	2.9626	14.9459	7.743	2.7757	1.0772
	IR H-MIMO	$\mathbf{1.8149 \times 10^{-6}}$	$\mathbf{2.3505 \times 10^{-7}}$	0.23985	0.98531	0.51268	**0.43857**
	ONBF	0.98313	0.9845	1.3339	3.8984	0.9663	1.3401

and 3 using polynomial basis functions can be more difficult as they do not belong to the class problem anymore. However, this is not a problem for the proposed method which does not require previous knowledge about the class problem.

Finally, note that if the cross-validation methodology were employed, the proposed method would be the one using the least number of points as no validation sets would be required.

9.5 Conclusions

In this chapter a new methodology for identifying MIMO Hammerstein Systems is presented where the impulse response of the system is incorporated into an LS-SVM formulation. This means that the model found benefits from the regularization capabilities of LS-SVM. It is shown that the proposed method is robust against the type of noise employed as even in the presence of high levels of noise it has a comparatively good performance, in fact, for the examples presented it performed better than the other methods considered.

The method proposed is very flexible with regard to the number of inputs and outputs it can handle and it is shown that it can model both SISO and MIMO systems. In addition, it has very good generalization capabilities and can work with different classes of problem. This constitutes a nice advantage when the class of problem is unknown or is difficult to model with certain basis functions. It is interesting to note that even though $\tilde{\boldsymbol{P}}$ is chosen arbitrarily the results are largely unchanged. This allows for certain flexibility when modeling the system.

Information from the structure of the system is included into the LS-SVM formulation allowing for an improvement of the modeling capabilities of LS-SVM for this type of systems. Nevertheless, the model is still obtained from a linear system of equations which represents an advantage.

Part IV

Additional Kernel Methods

In this part two works related to Machine Learning are presented. Both of them are of a general nature but can be specifically applied to system identification problems.

In Chapter 10 two methods are offered which allow a tradeoff between complexity and accuracy. It is shown there that with a small loss in accuracy, the complexity of the models can be greatly reduced. To do this, different versions of Fixed-Size Kernel models based on Fixed-Size Least Squares Support Vector Machines (FS-LSSVM) are employed. This chapter is based on the work presented in Castro, Mehrkanoon, Marconato, Schoukens, and Suykens (2014).

In Chapter 11 two new methods for extending the LS-SVM formulation to problems including dynamics are presented. In the first of these methods the frequency spectrum is divided into bands and a model focusing on each of such bands is estimated. Afterward all of these models are merged together. In the second one the procedure is similar, however, only a part of the spectrum is considered in this way. To complete the frequency spectrum, NARX LSSVM is used. These new methods are thoroughly compared with NARX-LSSVM in the context of real life data sets. This chapter is based on the work in Castro-Garcia, Tiels, and Suykens (2017).

Chapter 10

SVD truncation schemes for fixed-size kernel models

10.1 Introduction

When evaluating modeling techniques several performance criteria can be used. Normally, performance based on an error cost function is evaluated on a test set as this illustrates the generalization performance of the model. However, there might be other desirable characteristics of the models. For instance, where control is the goal of the identified model, a low complexity is also desirable by itself besides a good generalization capacity (Marconato, Schoukens, Rolain, & Schoukens, 2013).

For assessing the generalization performance of trained models without the use of validation data, various criteria have been developed. Such criteria take the general form of a prediction error (PE) which consists of the sum of two terms, namely PE = training error + complexity term. The complexity term represents a penalty growing with the number of free parameters in the model. Clearly, when the model is too simple it will be penalized by the residual error, but if it is too complex, it will be penalized by the complexity term. The minimum value for the criterion is given by a trade-off between the two terms (Bishop, 1995).

In Moody (1991) Moody generalized such criteria to deal with non-linear models and

This chapter is based on:
Castro, R., Mehrkanoon, S., Marconato, A., Schoukens, J., Suykens, J. (2014). *SVD truncation schemes for fixed-size kernel models*. In proceedings of the International Joint Conference on Neural Networks. IJCNN 2014. Beijing, China, Jun. 2014 (pp. 3922-3929).

to allow for the presence of a regularization term through the generalized prediction error which includes the effective number of parameters. Other approaches, like the one presented by Vapnik and Chervonenkis in V. N. Vapnik (1998) proposed an upper bound on the generalization error with a complexity term depending on the Vapnik-Chervonenkis dimension. Several other different theories with different notions of model complexity have been proposed in literature.

It is well-known that when applying regularization, instead of the number of parameters, the effective number of parameters is a more suitable notion then for model complexity. Also within support vector machines and kernel-based models the use of regularization is common (Suykens et al., 2002). Within the context of this chapter we will consider different versions of fixed-size kernel models related to fixed-size least squares support vector machines (Suykens et al., 2002). We will consider the Effective Degrees of Freedom (EDF), which are characterized by the trace of the hat matrix, as the notion for model complexity. The studied fixed-size kernel models relate to applying ordinary least squares and ridge regression in the primal, after obtaining a Nyström approximated feature map based on a selected subset of the given data. The resulting kernel models are sparse and the terminology of support vectors is used here for the Rényi based selected subset of prototype vectors. The size of the subset controls the degree of sparsity of the fixed-size kernel model.

Through this work, SVD truncation schemes for the fixed-size kernel models are investigated. It will be illustrated that even though these truncation schemes are not suited to further improve the generalization performance, the effective degrees of freedom can be greatly reduced. This realizes a reduction of the complexity of the resulting models and in this way, the resulting model can keep a fairly good generalization performance while at the same time getting a reduced complexity.

This chapter is organized as follows: In section 10.2, the SVD truncation schemes employed are presented and the concept of effective degrees of freedom is explained. Also, some practical considerations for the implementation done are exposed. In section 10.3 the Silverbox and Wiener-Hammerstein data sets are presented and the results found for the application of SVD truncation schemes are illustrated. These results are discussed on section 10.4. Finally, in section 10.5 the conclusions are given.

10.2 SVD truncation schemes

In Appendix B a review of Fixed Size LS-SVM is presented including the estimation of the feature map $\hat{\varphi}$. In Appendix C the EDF concept for LS-SVM is introduced as well.

Once $\hat{\varphi}$ is calculated, the model in primal form is computed according to the techniques described in this section. The particular studied estimation techniques are introduced in the present section as well as the effective degrees of freedom (*EDF*) for Fixed-Size Ordinary Least Squares (FS-OLS) and Fixed-Size Ridge Regression (FS-RR).

10.2.1 FS-OLS with truncation

After obtaining the optimal M subsample values through Quadratic Rényi Entropy (see Appendix B), the training points are projected into the feature space. This projection depends on the dimensionality given by the number of support vectors selected by the user (i.e. M)

$$\hat{\boldsymbol{\Phi}} = [\hat{\varphi}(\boldsymbol{x}_1), ..., \hat{\varphi}(\boldsymbol{x}_{N_{train}})]^\top \tag{10.1}$$

with $\boldsymbol{X}_{train} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{N_{train}}]$. From this, matrix $\boldsymbol{Q}$ is defined:

$$\boldsymbol{Q} = \hat{\boldsymbol{\Phi}}^\top \hat{\boldsymbol{\Phi}}. \tag{10.2}$$

The $\boldsymbol{Q}$ matrix can be decomposed through SVD resulting in $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$. Given that $\boldsymbol{Q}$ is a positive semi-definite matrix and $\boldsymbol{Q} = \hat{\boldsymbol{\Phi}}^\top \hat{\boldsymbol{\Phi}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ with $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}$, $\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}$ and $\boldsymbol{S}$ a diagonal matrix with positive diagonal elements, one has $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top = \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^\top$.

After decomposing $\boldsymbol{Q}$, the less relevant singular values from $\boldsymbol{S}$ are discarded successively and the reconstructed $\boldsymbol{Q}$ matrix $\hat{\boldsymbol{Q}} = \boldsymbol{U}\hat{\boldsymbol{S}}\boldsymbol{V}^\top$ is used in the validation set to determine the best truncation (i.e. how many singular values are discarded).

The *FS-OLS* model estimate with truncation becomes then:

$$\boldsymbol{w}_{OLS_{trun}} = \left(\boldsymbol{U}\hat{\boldsymbol{S}}\boldsymbol{V}^\top\right)^{-1} \hat{\boldsymbol{\Phi}}^\top \boldsymbol{y}_{train}. \tag{10.3}$$

Similarly to equation (10.1):

$$\hat{\boldsymbol{\Phi}}_{val} = \left[\hat{\varphi}(\boldsymbol{x}_1^{val}), ..., \hat{\varphi}(\boldsymbol{x}_{N_{val}}^{val})\right]^\top \tag{10.4}$$

with $\boldsymbol{X}_{val} = [\boldsymbol{x}_1^{val}, ..., \boldsymbol{x}_{N_{val}}^{val}]$. Therefore:

$$\hat{\boldsymbol{y}}_{val_{OLS,trun}} = \hat{\boldsymbol{\Phi}}_{val} \boldsymbol{w}_{OLS_{trun}}. \tag{10.5}$$

Once the best truncation is found, the system is applied to the test set:

$$\hat{\boldsymbol{y}}_{test_{OLS,trun}} = \hat{\boldsymbol{\Phi}}_{test} \boldsymbol{w}_{OLS_{trun}}. \tag{10.6}$$

Here, $\hat{\boldsymbol{\Phi}}_{test}$ is defined as:

$$\hat{\boldsymbol{\Phi}}_{test} = \left[\hat{\varphi}(\boldsymbol{x}_1^{test}), ..., \hat{\varphi}(\boldsymbol{x}_{N_{test}}^{test})\right]^\top \tag{10.7}$$

with $\boldsymbol{X}_{test} = [\boldsymbol{x}_1^{test}, ..., \boldsymbol{x}_{N_{test}}^{test}]$.

10.2.2 FS-RR with truncation

For ridge regression, $\hat{\boldsymbol{\Phi}}$, $\hat{\boldsymbol{\Phi}}_{val}$, $\hat{\boldsymbol{\Phi}}_{test}$ and $\boldsymbol{Q}$ are calculated in the same way as described in the *FS-OLS* method. However, the formulation changes as follows:

$$\begin{aligned} \boldsymbol{w}_{RR} &= \left(\hat{\boldsymbol{\Phi}}^\top \hat{\boldsymbol{\Phi}} + \lambda \boldsymbol{I}\right)^{-1} \hat{\boldsymbol{\Phi}}^\top \boldsymbol{y}_{train} \\ &= (\boldsymbol{Q} + \lambda \boldsymbol{I})^{-1} \hat{\boldsymbol{\Phi}}^\top \boldsymbol{y}_{train} \end{aligned} \tag{10.8}$$

where λ is the regularization parameter. Truncation of this solution becomes:

$$\begin{aligned} \boldsymbol{w}_{RR_{trun}} &= (\boldsymbol{U}\hat{\boldsymbol{S}}\boldsymbol{U}^\top + \lambda \boldsymbol{U}\boldsymbol{U}^\top)^{-1} \hat{\boldsymbol{\Phi}}^\top \boldsymbol{y}_{train} \\ &= \boldsymbol{U}\left(\hat{\boldsymbol{S}} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{U}^\top \hat{\boldsymbol{\Phi}}^\top \boldsymbol{y}_{train}. \end{aligned} \tag{10.9}$$

Once again, the most appropriate λ value is determined by the validation set (i.e. through *linesearch*) and finally, the resulting model is tested on the test set:

$$\hat{\boldsymbol{y}}_{val_{RR,trun}} = \hat{\boldsymbol{\Phi}}_{val} \boldsymbol{w}_{RR_{trun}} \tag{10.10}$$

and

$$\hat{\boldsymbol{y}}_{test_{RR,trun}} = \hat{\boldsymbol{\Phi}}_{test} \boldsymbol{w}_{RR_{trun}}. \tag{10.11}$$

For truncation, the same procedure is used as in FS-OLS, however, besides looking for the best λ value, also the best truncation is looked for. This results in a *gridsearch* approach.

10.2.3 SVD truncation as regularization

It is interesting to note that the truncation of the SVD can be seen as a regularization procedure by itself. To see this, first consider

$$\boldsymbol{w}_{RR} = (\hat{\boldsymbol{\Phi}}^\top \hat{\boldsymbol{\Phi}} + \lambda \boldsymbol{I})^{-1} \hat{\boldsymbol{\Phi}}^\top \boldsymbol{y}_{train} = \boldsymbol{M}\boldsymbol{y}_{train}. \tag{10.12}$$

With $\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Sigma}_M \boldsymbol{U}^\top$,

$$\boldsymbol{\Sigma}_M = \text{diag}\left(\frac{\sigma_1}{\sigma_1^2 + \lambda}, \ldots, \frac{\sigma_N}{\sigma_N^2 + \lambda}\right). \tag{10.13}$$

The regularization process can be seen then as a filtering of the contributions from the smallest singular values to the solution. This can further be seen then as a rewriting of the i^{th} element of $\boldsymbol{\Sigma}_M$ as the corresponding element of the original $\boldsymbol{S}$ matrix times a factor f_i (Hansen, 1990).

In the case of the truncation of the SVD, where the last singular value considered is σ_k:

$$f_i = \begin{cases} 1 & \text{for} \quad \sigma_i \quad \geq \quad \sigma_k \\ 0 & \text{for} \quad \sigma_i \quad < \quad \sigma_k \end{cases} . \tag{10.14}$$

Clearly, this corresponds to a sharp filter that cuts off the last $N - k$ singular values.

In the case of regularization the filter becomes instead

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}, \text{ for } i = 1, \ldots, N, \tag{10.15}$$

which in turn corresponds to a smooth filter dampening the components corresponding to $\sigma_i < \lambda$. This means then

$$\boldsymbol{w}_{RR} = \sum_{i=1}^{N} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \frac{\boldsymbol{u}_{(i)}^\top \boldsymbol{y}_{Train}}{\sigma_i} \boldsymbol{v}_{(i)}, \tag{10.16}$$

with $\boldsymbol{u}_{(i)}$ and $\boldsymbol{v}_{(i)}$ the i^{th} columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively.

Interestingly, if k is such that $\sigma_k = \lambda$, the sharp filter of the SVD truncation can be seen as an approximation to the smooth filter of the regularization.

10.2.4 Effective degrees of freedom

The hat matrix (see Appendix C), from where the effective degrees of freedom can be estimated (De Brabanter, De Brabanter, Suykens, & De Moor, 2011a), becomes for OLS:

$$\begin{aligned} \boldsymbol{H}_{OLS} &= \hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^\top \hat{\boldsymbol{\Phi}})^{-1} \hat{\boldsymbol{\Phi}}^\top \\ \boldsymbol{H}_{OLS_{trun}} &= \hat{\boldsymbol{\Phi}}(\boldsymbol{U}\hat{\boldsymbol{S}}^{-1}\boldsymbol{V}^\top)\hat{\boldsymbol{\Phi}}^\top. \end{aligned} \tag{10.17}$$

Similarly, for Ridge Regression and its truncated version:

$$\begin{aligned} \boldsymbol{H}_{RR} &= \hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^\top \hat{\boldsymbol{\Phi}} + \lambda \boldsymbol{I})^{-1} \hat{\boldsymbol{\Phi}}^\top \\ \boldsymbol{H}_{RR_{trun}} &= \hat{\boldsymbol{\Phi}}\boldsymbol{U}(\hat{\boldsymbol{S}} + \lambda \boldsymbol{I})^{-1}\boldsymbol{U}^\top \hat{\boldsymbol{\Phi}}^\top. \end{aligned} \tag{10.18}$$

10.3 Experimental results

In *FS-LSSVM* it is necessary to specify a subset of M input points to represent the data set reasonably well. For this purpose, the quadratic Rényi entropy is used and an approximation to the feature map is calculated as explained in Appendix B. A *gridsearch* approach is used then to tune the values of the tuning parameters λ and σ.

The parameters are selected in accordance with the results obtained from evaluating the resulting model on the validation set. The chosen model is finally used on the test set.

Note that the structure of the model will be that of a nonlinear autoregressive model with exogenous input (NARX), where the model relates the current value of a time series with past values of the same series and current and past values of the driving (exogenous) series. A NARX model can be expressed as follows:

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, u_t, u_{t-1}, u_{t-2}, \ldots, u_{t-p}) \tag{10.19}$$

where $f(\cdot)$ is some nonlinear function and $\hat{y}_t$ is the estimated value of y. Here y is the variable of interest, u is the external input and p is the number of lags used determining how many past u and y values are included to calculate $\hat{y}$.

In this section, the results obtained by applying the techniques presented in this chapter, under the one-step ahead framework, are presented. Also, a description of the data sets used is offered.

10.3.1 Silverbox data set

The Silverbox data set was introduced in J. Schoukens, Nemeth, Crama, Rolain, and Pintelon (2003). This data set represents an electrical circuit simulating a mass-spring damper system. It is a nonlinear dynamic system with feedback exposing a dominant linear behavior (Espinoza et al., 2004).

In Figure 10.1, the inputs and outputs of the system are depicted. The data set consists of 131072 data points and was split evenly between test, validation and training sets.

10.3.2 Wiener-Hammerstein data set

The concatenation of two linear systems with a static nonlinearity in between constitutes an important special class of nonlinear systems known as a Wiener-Hammerstein system (Giri & Bai, 2010).

The Wiener-Hammerstein data set[1] was introduced in J. Schoukens et al. (2009). The system modelled is an electronic nonlinear system with a Wiener-Hammerstein structure as shown in Figure 10.2. There, G_1 is a third order Chebyshev filter, G_2 is a third order inverse Chebyshev filter and the static nonlinearity is built using a diode circuit. The measured input and output of the circuit are as shown on Figure 10.3. The data set consists of 188000 data points and was split evenly between test, validation and training sets.

[1] *http://tc.ifac-control.org/1/1/Data%20Repository/sysid-2009-wiener-hammerstein-benchmark*

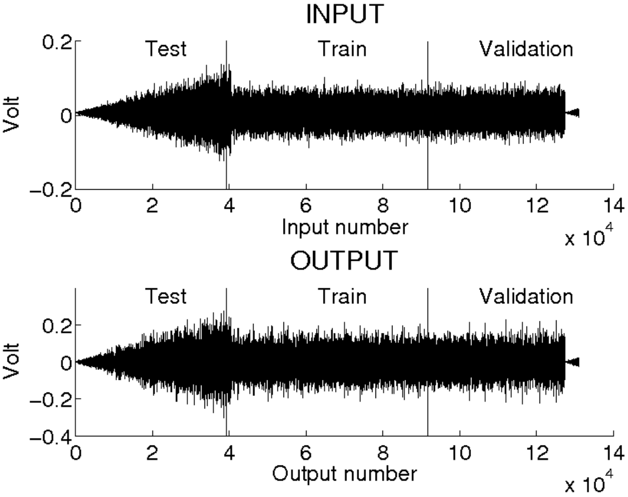


Figure 10.1: Silverbox benchmark data set

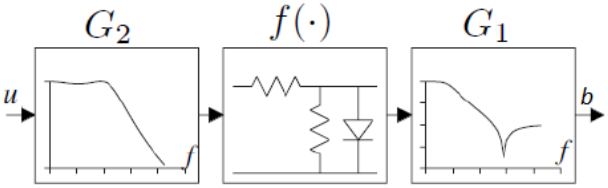


Figure 10.2: Taken from J. Schoukens et al. (2009). Wiener-Hammerstein system consisting of a linear dynamic block G_1, a static non-linear block $f(\cdot)$ and a linear dynamic block G_2

10.3.3 Truncation and generalization performance

When the systems described in Section 10.2 are subjected to truncation, the general result obtained on the data sets used in this work is that the generalization performance decreases. This implies that if only the generalization performance is considered, the models should either remain unchanged or the truncation should be very minor in order to avoid the decrease in generalization performance. However, if a compromise between generalization performance and complexity is allowed, the situation changes dramatically. This can be seen in Figures 10.4 and 10.5 where a 10% in decreased generalization performance is allowed (i.e. the best generalization performance value is multiplied by 1.1 and this value is used as a tolerance threshold).

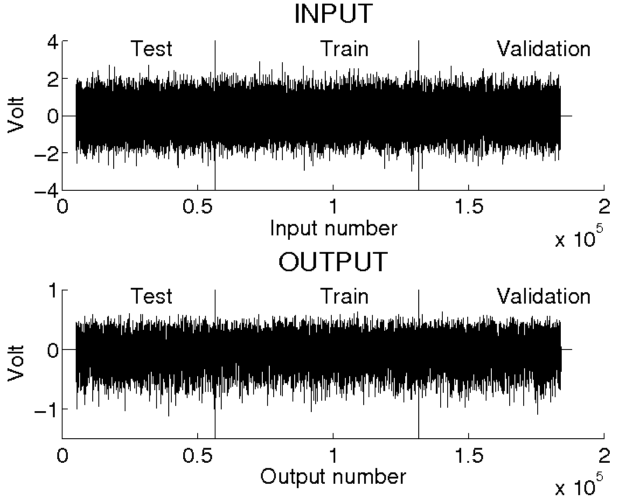


Figure 10.3: Wiener-Hammerstein benchmark data set

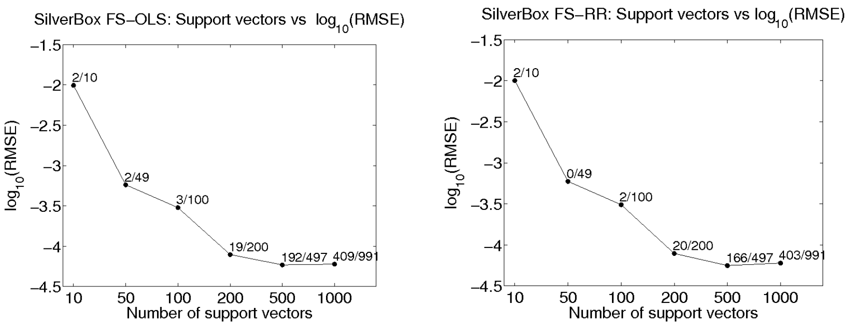


Figure 10.4: Test set performance vs. Number of support vectors on the Silverbox benchmark data set. At each point the relation between the number of singular values truncated and the total number of singular values is displayed.

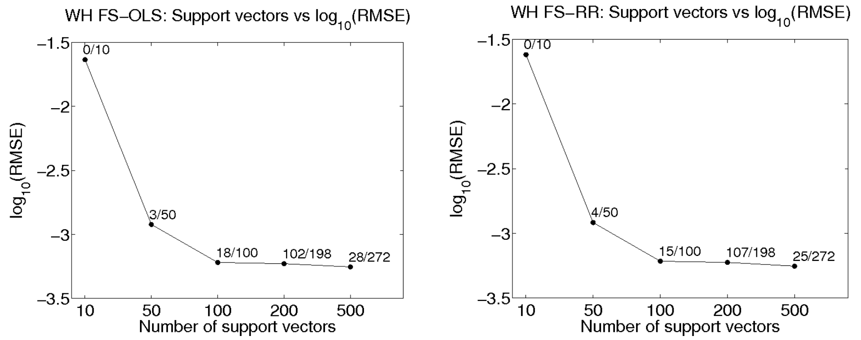


Figure 10.5: Test set performance vs. Number of support vectors on the Wiener-Hammerstein benchmark data set. At each point the relation between the number of singular values truncated and the total number of singular values is displayed.

In Figures 10.6 and 10.7 the resulting selection (i.e. with the 10% threshold) is represented by the diamond shaped markers. As can be seen, the more support values the system uses, the greater the reduction of singular values that can be achieved.

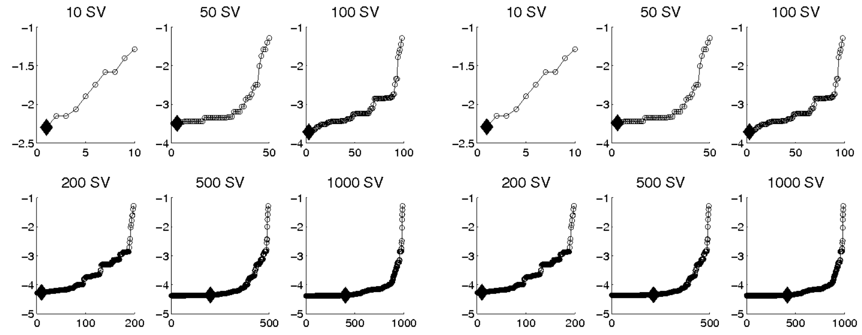


Figure 10.6: Compromise of up to 10% of test set performance for reduced complexity in the Silverbox benchmark data set. Horizontal axis represents the number of Singular Values eliminated. Vertical axis represents the test performance ($\log_{10}(RMSE)$). (Left) FS-RR. (Rigth) FS-OLS.

Note that this holds for both data sets and for both FS-OLS and FS-RR methods. This behavior already suggests that the effective degrees of freedom can be greatly reduced if a small compromise of the generalization performance is allowed. This idea will be developed in section 10.3.4.

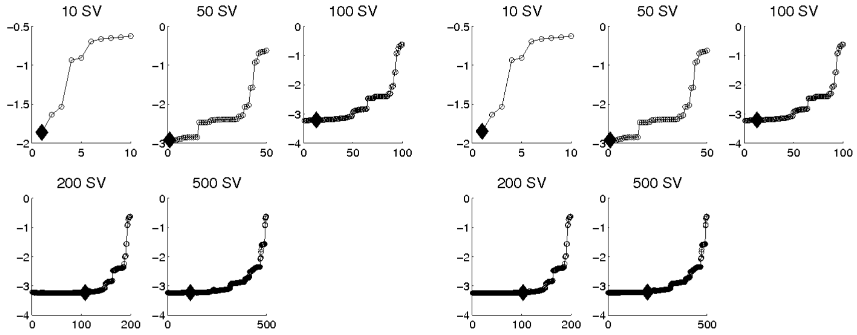


Figure 10.7: Compromise of up to 10% of test set performance for reduced complexity in the Wiener-Hammerstein benchmark data set. Horizontal axis represents the number of Singular Values eliminated. Vertical axis represents the test performance ($\log_{10}(RMSE)$). (Left) FS-RR. (Rigth) FS-OLS.

10.3.4 Effective number of parameters

The definitions in section 10.2.4 allow the representation of the effective degrees of freedom (given the different possible truncations) versus the generalization performance of the model. Figures 10.8 and 10.9 illustrate these results. Note that in this case, not only a good generalization performance is desired, but also a model with a reduced complexity. A compromise between both of them must be achieved. The lines suggest a possibly good choice for this compromise. To draw these lines, the axes are rescaled so they have the same scale and the point with the minimum combined distance to the vertical axis and the lowest error in the rescaled axes is chosen. The rescaling is done to give the same relevance to both axes. The line is then drawn with the axes in their original scale and the graphs show that in these cases, it is possible indeed to greatly reduce the effective degrees of freedom without much loss of generalization performance.

10.4 Discussion

It has been shown in section 10.3.3 that when applying SVD truncation schemes for Fixed-Size kernel models, in principle a significant reduction of support vectors is not to be expected if the generalization performance is to be maximized. However, if a trade-off between generalization performance and complexity is allowed, a significant truncation of the singular values of the Q matrix can be made. Furthermore, it has been

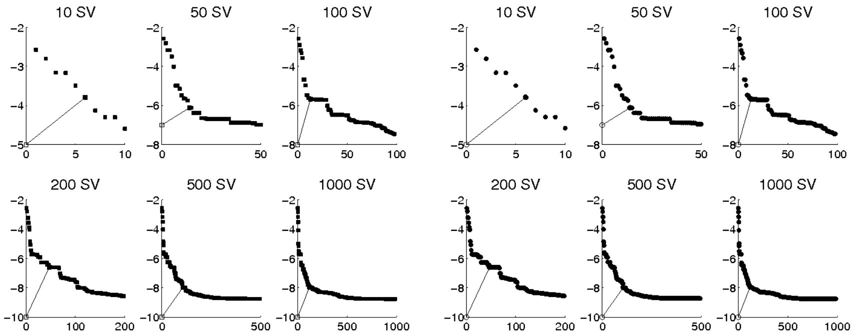


Figure 10.8: Test set performance vs EDF on the Silverbox benchmark data set for different fixed sizes. Horizontal axes represent the number of remaining effective degrees of freedom after truncation (i.e. $tr(H)$). The vertical axes represent the test set performance ($\log_{10}(RMSE)$). (Left) FS-RR. (Rigth) FS-OLS.

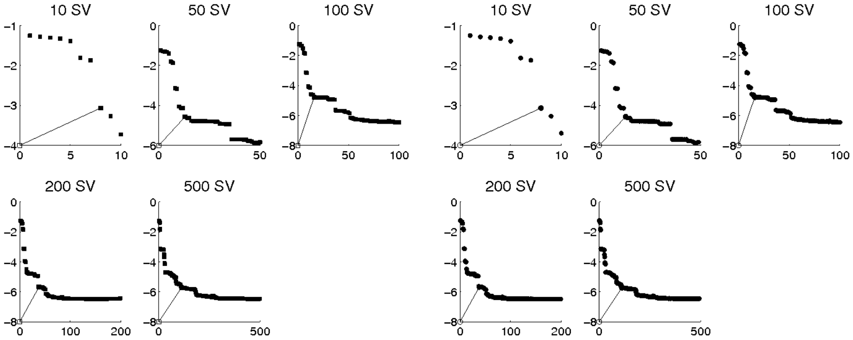


Figure 10.9: Test set performance vs EDF for RR on the Wiener-Hammerstein benchmark data set for different fixed sizes. Horizontal axes represent the number of remaining effective degrees of freedom after truncation (i.e. $tr(H)$). The vertical axes represent the test set performance ($\log_{10}(RMSE)$). (Left) FS-RR. (Rigth) FS-OLS.

shown that the complexity of the system, in terms of the effective degrees of freedom, can be greatly reduced through singular value truncation without a big impact on the generalization performance.

The results presented are relevant as they demonstrate that when employing Fixed-Size kernel models, it is possible to obtain models with highly reduced complexity when SVD truncation schemes are applied. However, those models will have a small reduction on generalization performance. This is desirable when the identified model is used e.g. for control purposes and when parsimonious models are preferred (Ljung, 1999; Marconato et al., 2013).

These findings are in line with Marconato et al. (2013), Moody (1991) and Spiegelhalter, Best, Carlin, and Van Der Linde (2002) as they illustrate that indeed the effective degrees of freedom for a Fixed-Size kernel model can greatly differ from the number of parameters of the system. In other words, the effective degrees of freedom can be much smaller than the number of support vectors in the Fixed-Size models.

10.5 Conclusions

In this chapter we have considered different truncation schemes for fixed-size Kernel models based on SVD. It has been shown that if a compromise between generalization performance and complexity is allowed, the effective degrees of freedom of the underlying system can be greatly reduced on Fixed-Size kernel models without much loss of generalization performance.

FS-OLS and FS-RR methods have shown to very efficiently reduce the effective degrees of freedom of Fixed-Size kernel models under an SVD truncation scheme. In fact, the methods presented have been successfully applied on two well-known benchmark data sets in system identification: the Wiener-Hammerstein and Silverbox data sets where similar and consistent results were obtained. Possible future work may explore related methods for other possible model structures.

Chapter 11

Frequency Division Least Squares Support Vector Machines

11.1 Introduction

A frequency-domain formulation of LS-SVM is presented in Lataire, Piga, and Tóth (2014) and Lataire, Pintelon, Piga, and Tóth (2017). In these works the focus is on estimating discrete-time and continuous-time linear time-varying (LTV) systems, respectively. The frequency-domain formulation can deal with stationary correlated (over time) noise, since stationary noise is not correlated over the frequency. Also, the frequency-domain formulation has the added advantage that it is easy to focus on a frequency band of interest (Lataire et al., 2017).

The methods proposed in this chapter introduce a frequency-domain weighting in a time-domain formulation of LS-SVM for the estimation of nonlinear systems using a NARX model structure. Since the transformation from the time domain to the frequency domain is a linear orthonormal transformation given by the DFT matrix, the proposed method fits in the setting of weighted LS-SVM (Suykens et al., 2002), with a particular choice for the weights. However, an initial estimation of the weights using an unweighted LS-SVM is not needed as they follow from the selection of a frequency

This chapter is based on:
Castro-Garcia, R., Tiels, K., Suykens, J. A. K. (2017). *Frequency Division LS-SVM for Nonlinear Modeling*. In Internal report 17-24, ESAT-SISTA, KU Leuven (Leuven, Belgium).

band of interest. Moreover, the computational efficiency of LS-SVM is maintained due to the linear orthonormal transformation to go from the time to the frequency domain.

The proposed method is illustrated on several datasets from DaISy (De Moor, De Gersem, De Schutter, & Favoreel, 1997), which is a collection of datasets for the identification of systems. For each dataset, the output spectrum is divided in a number of frequency bands using the frequency weighting. An LS-SVM model is trained for each particular frequency band and the overall model is obtained as the sum of the individual models. This technique can be seen as a local nonlinear modeling where each local model takes care of a certain frequency band. As such, the method is different from local modeling approaches as in (Münker & Nelles, 2016) where local FIR or ARX linear models are constructed and combined using validity functions.

Applications where data are split over several frequency bands and a specific LS-SVM classification model is trained based on the best suited frequency band are reported in Jiang, Liu, and He (2012) and Xie, Wang, and He (2014). A similar approach, but with a regression goal in mind, is reported in Li, Xie, He, Qiu, and Zhang (2012).

This chapter is organized as follows: In Section 11.2 the proposed methods are presented. Section 11.3 illustrates the results found when applying the described methodology to several real life and simulation examples. Finally, in Section 11.4, the conclusions are presented.

11.2 Proposed methods

Two methodologies are proposed that allow a better performance than the standard NARX LS-SVM. In particular, these methods are well suited to deal with nonlinear dynamical systems which usually pose a difficult modeling problem.

11.2.1 Frequency Division LSSVM

As explained in Section 1.7.1, the framework of LS-SVM is given by a primal-dual formulation. For the methods proposed in this chapter that framework is kept. Given the data set $\{\boldsymbol{x}(t), y(t)\}_{t=1}^{N}$, the objective is to find a model

$$\hat{y}(t) = \boldsymbol{w}^\top \varphi(\boldsymbol{z}(t)) + b \tag{11.1}$$

where $\boldsymbol{z}(t) = [y(t-1), y(t-2), \ldots, y(t-r_a), \boldsymbol{x}^\top(t), \boldsymbol{x}^\top(t-1), \ldots, \boldsymbol{x}^\top(t-r_b)]$ with $\boldsymbol{z}_k \in \mathbb{R}^{r_a+n(r_b+1)}$, $\boldsymbol{w}$ is a weight vector, $\boldsymbol{x}(t-i) \in \mathbb{R}^n$ with $i = 0, \ldots, r_b$ are the present and past inputs, $y(t-i) \in \mathbb{R}$, with $i = 1, \ldots, r_a$ are the past outputs and $\hat{y}(t) \in \mathbb{R}$ denotes the estimated current output. Also, $\varphi(\cdot) : \mathbb{R}^{r_a+n(r_b+1)} \to \mathbb{R}^{n_h}$ is the feature map to a high dimensional (possibly infinite) space and b is the bias term.

It is possible to formulate an optimization problem where emphasis is given specifically to a frequency band. To do this let us define a discrete Fourier transform matrix as

$$\boldsymbol{A} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{2\pi \mathrm{j}}{N}} & e^{-\frac{4\pi \mathrm{j}}{N}} & \cdots & e^{-\frac{2(N-1)\pi \mathrm{j}}{N}} \\ 1 & e^{-\frac{4\pi \mathrm{j}}{N}} & e^{-\frac{8\pi \mathrm{j}}{N}} & \cdots & e^{-\frac{4\pi \mathrm{j}}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2(N-1)\pi \mathrm{j}}{N}} & e^{-\frac{4(N-1)\pi \mathrm{j}}{N}} & \cdots & e^{-\frac{2(N-1)^2\pi \mathrm{j}}{N}} \end{bmatrix}, \tag{11.2}$$

with $\mathrm{j} = \sqrt{-1}$.

This means that it is possible to retrieve a representation of the error vector $\boldsymbol{e}$ (i.e. see (1.29)) in the frequency domain as

$$\mathcal{F}(\boldsymbol{e}) = \boldsymbol{A}\boldsymbol{e}, \tag{11.3}$$

where the Fourier transform operator $\mathcal{F}$ is such that $\mathcal{F}(x(t)) = X(k)$ and $\mathcal{F}^{-1}(X(k)) = x(t)$[1].

Given that the intention is to focus only on one specific frequency band, a diagonal matrix $\boldsymbol{P}$ is created. For an emphasis in the frequency band between frequencies f_A and f_B, $\boldsymbol{P}$ will have a diagonal $\boldsymbol{p}$ of the form

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \\ \boldsymbol{p}_3 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_1 \end{bmatrix}, \tag{11.4}$$

with $\boldsymbol{p}_1 = \boldsymbol{0}_{n_1}$, $\boldsymbol{p}_2 = \boldsymbol{1}_{n_2}$ and $\boldsymbol{p}_3 = \boldsymbol{0}_{n_3}$. Here n_1 represents the number of bins between 0Hz and f_A, n_2 those between f_A and f_B and n_3 those between f_B and $\frac{f_s}{2}$ with f_s the sampling frequency. Fig. 11.1 illustrates an example of the diagonal of $\boldsymbol{P}$.

An optimization problem is then formulated:

$$\begin{aligned} \min_{\boldsymbol{w},b,e} \quad & \tfrac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \tfrac{\gamma}{2}\boldsymbol{e}^\top \boldsymbol{e} + \tfrac{\eta}{2}\boldsymbol{e}^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{e} \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{\Phi}^\top \boldsymbol{w} + \boldsymbol{1}_N b + \boldsymbol{e}, \end{aligned} \tag{11.5}$$

where γ and η are regularization parameters, $\boldsymbol{\Phi} = [\varphi(\boldsymbol{z}_1), \varphi(\boldsymbol{z}_2), \ldots, \varphi(\boldsymbol{z}_N)] \in \mathbb{R}^{n_h \times N}$ and $\varphi(\cdot) : \mathbb{R}^{r_a + n(r_b+1)} \to \mathbb{R}^{n_h}$. Note that the term $\frac{\eta}{2}\boldsymbol{e}^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}\boldsymbol{e}$ will force the model to specially focus in the frequency band defined by $\boldsymbol{P}$. Also, note that

[1] In this work it is assumed that $\mathcal{F}(x(t)) \in \mathbb{C}$ and $\mathcal{F}^{-1}(X(k)) \in \mathbb{R}$

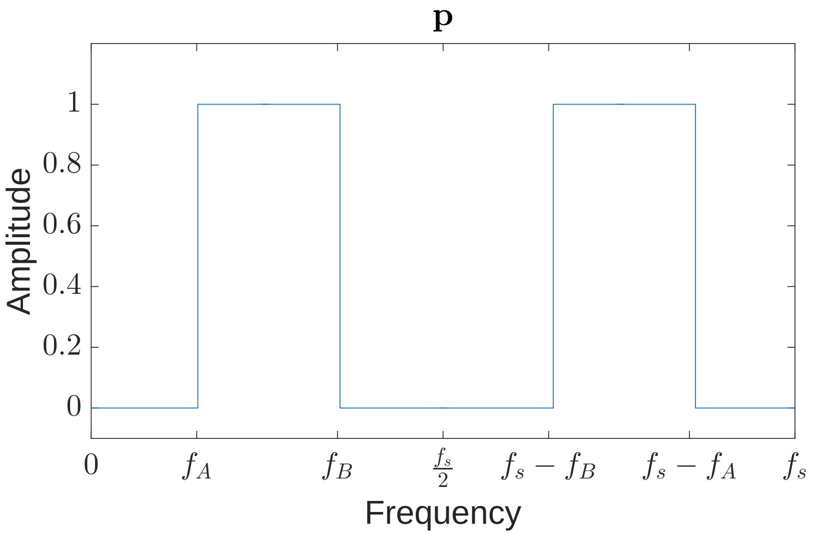


Figure 11.1: Example of the diagonal of matrix $\boldsymbol{P}$ used to emphaize a frequency band between f_A and f_B.

the term $\frac{\gamma}{2}\boldsymbol{e}^\top\boldsymbol{e}$ could be replaced by $\frac{\gamma}{2}\boldsymbol{e}^\top\boldsymbol{\Psi}\boldsymbol{e}$ where $\boldsymbol{\Psi}$ would be a weighting matrix. However, for the remaining of this chapter we will have $\boldsymbol{\Psi} = \boldsymbol{I}$. Finally, we will assume that the term $\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{A} \in \mathbb{R}$.

Through the use of Mercer's theorem (Mercer, 1909), the entries of the kernel matrix can be represented by $\boldsymbol{\Omega}_{i,j} = \varphi(\boldsymbol{x}_i)^\top\varphi(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with $i, j = 1, ..., N$ (and therefore $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$). The function $\varphi(\cdot)$ does not have to be explicitly known then as it is used implicitly through the positive definite kernel function. In this chapter, the radial basis function kernel (RBF kernel) is used i.e. $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2 / \sigma^2)$ where σ is the kernel parameter.

The Lagrangian is then formulated

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, e; \alpha) = \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\gamma}{2}\boldsymbol{e}^\top\boldsymbol{e} + \frac{\eta}{2}\boldsymbol{e}^\top\boldsymbol{A}^\top\boldsymbol{P}\boldsymbol{A}\boldsymbol{e} \\ -\boldsymbol{\alpha}^\top(\boldsymbol{\Phi}^\top\boldsymbol{w} + \boldsymbol{1}_N b + \boldsymbol{e} - \boldsymbol{y}),\end{aligned} \tag{11.6}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^N$ the Lagrange multipliers. The optimality conditions for this formulation are then:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 & \rightarrow \quad \boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow \quad \mathbf{1}_N^\top \boldsymbol{\alpha} = 0 \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{e}} = 0 & \rightarrow \quad 0 = \gamma \boldsymbol{e}^\top + \eta \boldsymbol{e}^\top \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A} - \boldsymbol{\alpha}^\top \\ & \qquad \boldsymbol{\alpha}^\top = \boldsymbol{e}^\top (\boldsymbol{I}_N \gamma + \eta \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A}) \\ & \qquad \boldsymbol{e} = ((\gamma \boldsymbol{I}_N + \eta \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A})^{-1})^\top \boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \rightarrow \quad \boldsymbol{y} = \boldsymbol{\Phi}^\top \boldsymbol{w} + \mathbf{1}_N b + \boldsymbol{e}. \end{cases} \tag{11.7}$$

By elimination of $\boldsymbol{w}$ and $\boldsymbol{e}$ the last equation in 11.7 can be rewriten as

$$\boldsymbol{y} = \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\boldsymbol{\alpha} + \mathbf{1}_N b + ((\gamma \boldsymbol{I}_N + \eta \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A})^{-1})^\top \boldsymbol{\alpha}. \tag{11.8}$$

The following linear system is then obtained

$$\begin{bmatrix} 0 & \mathbf{1}_N^\top \\ \mathbf{1}_N & \boldsymbol{\Omega} + (\gamma \boldsymbol{I}_N + \eta \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{A})^{-1} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}, \tag{11.9}$$

and the resulting model for a new set is

$$\begin{aligned} \hat{\boldsymbol{y}}_{new} &= \boldsymbol{\Phi}_{New}^\top \boldsymbol{w} + \mathbf{1}_{N_{new}} b, \\ &= \boldsymbol{\Phi}_{New}^\top \boldsymbol{\Phi}\boldsymbol{\alpha} + \mathbf{1}_{N_{new}} b, \\ &= \boldsymbol{K}\boldsymbol{\alpha} + \mathbf{1}_{N_{new}} b, \end{aligned} \tag{11.10}$$

with $\boldsymbol{\Phi}_{New} = [\varphi(\boldsymbol{z}_{new,1}), \varphi(\boldsymbol{z}_{new,2}), \ldots, \varphi(\boldsymbol{z}_{new,N_{new}})] \in \mathbb{R}^{n_h \times N_{new}}$, N_{new} the number of points of the new data set and $\boldsymbol{K} \in \mathbb{R}^{N_{new} \times N}$ with $\boldsymbol{K}_{ij} = k(\boldsymbol{z}_{new,i}, \boldsymbol{z}_j)$ and $i = 1, \ldots, N_{new}$, $j = 1, \ldots, N$.

It is interesting to note that the application of the model is exactly of the same form as that of a standard LS-SVM. Furthermore, the model obtained in 11.9 is very similar to the standard one (i.e. see (1.32)). For instance if $\boldsymbol{P} = \boldsymbol{I}_N$, given that $\boldsymbol{A}$ is an orthogonal matrix, the resulting linear system would be

$$\begin{bmatrix} 0 & \mathbf{1}_N^\top \\ \mathbf{1}_N & \boldsymbol{\Omega} + ((\gamma \boldsymbol{I}_N + \eta \boldsymbol{I}_N)^{-1})^\top \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix} \tag{11.11}$$

and therefore

$$\begin{bmatrix} 0 & \mathbf{1}_N^\top \\ \mathbf{1}_N & \boldsymbol{\Omega} + \frac{\boldsymbol{I}_N}{\gamma + \eta} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}, \tag{11.12}$$

which is equivalent to the model in (1.32). This shows that for the method to work it is necessary that the frequency band considered is smaller than the whole frequency spectrum considered (i.e. from 0Hz to f_s). Otherwise, the model found will be equivalent to the NARX LSSVM method.

With the method presented so far, we can obtain a model capable of focusing in a particular frequency band given by $\boldsymbol{P}$. However, the procedure can be repeated in order to cover a broader range of the frequency spectrum. This means that for a given division of the spectrum where $p+1$ bands are selected, the method can retrieve a final model including all of the frequencies considered by merging together the found $p+1$ models $\mathcal{M}_i$ with $i = 1, \dots, p+1$ (i.e. one model for each of the frequency bands of interest).

Given the intrinsic focus on a certain frequency band of each of the models $\mathcal{M}_i$, it is necessary to take the estimation of each of them, for an input $\boldsymbol{U}_{test}$, to the frequency domain and filter such estimation using the corresponding $\boldsymbol{P}_i$. This is:

$$\hat{\boldsymbol{y}} = \mathcal{F}^{-1}\left(\sum_{i=1}^{p+1} \boldsymbol{P}_i \mathcal{F}(\hat{\boldsymbol{y}}_i)\right). \tag{11.13}$$

In Algorithm 6 a summary of the Frequency Division LS-SVM (FD-LSSVM) method is presented. In Section 11.3 an illustrative example is shown.

Algorithm 6 Frequency Division LS-SVM.

```
  Define the division vector $\boldsymbol{f} = [0, f_1, f_2, \dots, f_p, \frac{f_s}{2}] \in \mathbb{R}^{p+2}$ containing the
  frequencies determining the frequency bands required.
  for $i := 1$ to $p+1$ do
      Create $\boldsymbol{p}_i$ with frequencies $f_A = \boldsymbol{f}_i$ and $f_B = \boldsymbol{f}_{i+1}$ as shown in (11.4).
      Create matrix $\boldsymbol{P}_i = \text{diag}(\boldsymbol{p}_i)$.
      Use (11.9) to tune a model $\mathcal{M}_i$.
      Obtain $\hat{\boldsymbol{y}}_i$ through (11.10) using $\mathcal{M}_i$ and $\boldsymbol{U}_{test}$.
  Estimate $\hat{\boldsymbol{y}}$ using (11.13).
```

11.2.2 Effective degrees of freedom

The complexity analysis, in terms of the effective degrees of freedom (EDF), for FD-LSSVM is fairly similar to that of LS-SVM shown in Appendix C.

Single sub model

From (11.9) for a particular frequency band a model $\mathcal{M}_i$ is obtained

$$\boldsymbol{y}_{tr,i} = (\boldsymbol{\Omega}_i + (\gamma_i \boldsymbol{I} + \eta_i \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A})^{-1})\boldsymbol{\alpha}_i + \mathbf{1}_N b_i. \tag{11.14}$$

From (11.14), $\boldsymbol{\alpha}_i$ and b_i can be written as

$$\begin{cases} \boldsymbol{\alpha}_i &= (\boldsymbol{\Omega}_i + (\gamma_i \boldsymbol{I} + \eta_i \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A})^{-1})^{-1}(\boldsymbol{y}_{tr} - \mathbf{1}_N b_i), \\ b_i &= \dfrac{\mathbf{1}_N^\top (\boldsymbol{\Omega}_i + (\gamma_i \boldsymbol{I} + \eta_i \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A})^{-1})^{-1} \boldsymbol{y}_{tr}}{\mathbf{1}_N^\top (\boldsymbol{\Omega}_i + (\gamma_i \boldsymbol{I} + \eta_i \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A})^{-1})^{-1} \mathbf{1}_N}. \end{cases} \tag{11.15}$$

Let us define now

$$\begin{cases} c_i = \mathbf{1}_N^\top (\boldsymbol{\Omega}_i + (\gamma_i \boldsymbol{I} + \eta_i \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A})^{-1})^{-1} \mathbf{1}_N, \\ \boldsymbol{Z}_i = \boldsymbol{\Omega}_i + (\gamma_i \boldsymbol{I} + \eta_i \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A})^{-1}, \\ \boldsymbol{J} = \mathbf{1}_N \mathbf{1}_N^\top. \end{cases} \tag{11.16}$$

From (11.10), for the training set we have

$$\hat{\boldsymbol{y}}_{tr,i} = \boldsymbol{\Omega}_i \boldsymbol{\alpha}_i + \mathbf{1}_N b_i. \tag{11.17}$$

Therefore,

$$\begin{aligned} \hat{\boldsymbol{y}}_{tr,i} &= \left(\boldsymbol{\Omega}_i \left(\boldsymbol{Z}_i^{-1} - \boldsymbol{Z}_i^{-1} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) + \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) \boldsymbol{y}_{tr} \\ &= \boldsymbol{H}_i \boldsymbol{y}_{tr}. \end{aligned} \tag{11.18}$$

Combined models

From (11.13) and (11.18)

$$\hat{\boldsymbol{y}}_{tr} = \sum_{i=1}^{p+1} \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \left(\boldsymbol{\Omega}_i \left(\boldsymbol{Z}_i^{-1} - \boldsymbol{Z}_i^{-1} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) + \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) \boldsymbol{y}_{tr}, \tag{11.19}$$

and therefore

$$\boldsymbol{H} = \sum_{i=1}^{p+1} \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \left(\boldsymbol{\Omega}_i \left(\boldsymbol{Z}_i^{-1} - \boldsymbol{Z}_i^{-1} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) + \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) \tag{11.20}$$

The EDF are then

$$\begin{aligned} EDF &= \sum_{i=1}^{p+1} \operatorname{tr} \left(\boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \left(\boldsymbol{\Omega}_i \left(\boldsymbol{Z}_i^{-1} - \boldsymbol{Z}_i^{-1} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) + \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) \right) \\ &= \sum_{i=1}^{p+1} \operatorname{tr} \left(\boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \boldsymbol{\Omega}_i \boldsymbol{Z}_i^{-1} - \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \boldsymbol{\Omega}_i \boldsymbol{Z}_i^{-1} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} + \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \right) \\ &= \sum_{i=1}^{p+1} \operatorname{tr} \left(\boldsymbol{\Omega}_i \boldsymbol{Z}_i^{-1} \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} - \boldsymbol{\Omega}_i \boldsymbol{Z}_i^{-1} \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} + \frac{\boldsymbol{J}}{c_i} \boldsymbol{Z}_i^{-1} \boldsymbol{A}^\top \boldsymbol{P}_i \boldsymbol{A} \right) \end{aligned} \tag{11.21}$$

Note that

$$\begin{aligned}
\boldsymbol{Z}_i^{-1}\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A} &= \left(\boldsymbol{\Omega}_i + \left(\gamma_i\boldsymbol{I} + \eta_i\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A}\right)^{-1}\right)^{-1}\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A} \\
&= \left((\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A})^{-1}\boldsymbol{\Omega}_i + (\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A})^{-1}\left(\gamma_i\boldsymbol{I} + \eta_i\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A}\right)^{-1}\right)^{-1} \\
&= \left((\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A})^{-1}\boldsymbol{\Omega}_i + \left(\gamma_i(\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A}) + \eta_i\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A})\right)^{-1}\right)^{-1} \\
&= \left((\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A})^{-1}\boldsymbol{\Omega}_i + \left(\gamma_i(\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A}) + \eta_i\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A}\right)^{-1}\right)^{-1} \\
&= \left((\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A})^{-1}\left(\boldsymbol{\Omega}_i + (\gamma_i+\eta_i)^{-1}\boldsymbol{I}\right)\right)^{-1} \\
&= \left(\boldsymbol{\Omega}_i + (\gamma_i+\eta_i)^{-1}\boldsymbol{I}\right)^{-1}\boldsymbol{A}^\top\boldsymbol{P}_i\boldsymbol{A},
\end{aligned} \tag{11.22}$$

and from (11.22) in (11.21)

$$EDF = \sum_{i=1}^{p+1} \mathrm{tr}\left(\boldsymbol{A}\left(\boldsymbol{\Omega}_i\left(\boldsymbol{M}_i^{-1} - \boldsymbol{M}_i^{-1}\frac{\boldsymbol{J}}{c_i}\boldsymbol{M}_i^{-1}\right) + \frac{\boldsymbol{J}}{c_i}\boldsymbol{M}_i^{-1}\right)\boldsymbol{A}^\top\boldsymbol{P}_i\right), \tag{11.23}$$

with $\boldsymbol{M}_i = \boldsymbol{\Omega}_i + \dfrac{\boldsymbol{I}}{(\gamma_i+\eta_i)}$. It is important to highlight the similarity between $\boldsymbol{M}$ and the matrix $\boldsymbol{Z}$ described in (C.3) and to note that the matrix $\left(\boldsymbol{\Omega}_i\left(\boldsymbol{M}_i^{-1} - \boldsymbol{M}_i^{-1}\dfrac{\boldsymbol{J}}{c_i}\boldsymbol{M}_i^{-1}\right) + \dfrac{\boldsymbol{J}}{c_i}\boldsymbol{M}_i^{-1}\right)$ is very similar to the hat matrix of the standard LS-SVM model (i.e. see Appendix C and in particular (C.5)).

Under the assumption that $\mathcal{O}(\mathrm{tr}(\boldsymbol{H}_i)) = \mathcal{O}(\mathrm{tr}(\boldsymbol{H}_{LSSVM}))$ we can see that (11.23) can be rewritten as $EDF_{FD-LSSVM} = \sum_{i=1}^{p+1}\mathrm{tr}\left(\boldsymbol{P}_i\boldsymbol{A}\boldsymbol{H}_i\boldsymbol{A}^\top\boldsymbol{P}_i\right)$ and therefore $\mathcal{O}(EDF_{FD-LSSVM}) = \mathcal{O}(EDF_{LSSVM})$.

11.2.3 Partial FD-LSSVM

Given that for a division vector with $p+2$ frequencies $p+1$ models have to be tuned, FD-LSSVM can be computationally demanding. In this section we offer an alternative where a tradeoff between the processing time and the accuracy of the results takes place. This alternative, from now on referred to as Partial FD-LSSVM, combines standard NARX LS-SVM and FD-LSSVM to improve the results of the former without requiring the processing time of the later. Although its accuracy is not as good as that of FD-LSSVM, it still surpasses that of NARX LS-SVM.

The idea in this approach is that not all the frequency spectrum is covered by the frequency bands determined by the division vector. Such remaining frequencies are covered instead by the standard NARX LS-SVM. This implies that, following (11.13), Partial FD-LSSVM can be expressed as

$$\hat{\boldsymbol{y}} = \mathcal{F}^{-1}\left(\sum_{j=1}^{q-1} \boldsymbol{P}_j \mathcal{F}(\hat{\boldsymbol{y}}_j) + \boldsymbol{Q}_h \mathcal{F}(\hat{\boldsymbol{y}}_{NARX})\right), \tag{11.24}$$

where $\boldsymbol{Q}_h = \boldsymbol{I} - \sum_{j=1}^{q-1} \boldsymbol{P}_j$ and $\boldsymbol{Q}_h \neq \boldsymbol{0}$.

In Algorithm 7 a summary of the Partial Frequency Division LS-SVM method is presented.

Algorithm 7 Partial Frequency Division LS-SVM.

```
Estimate $\hat{\boldsymbol{y}}_{NARX}$ as described in Section 1.7.2.
Define the division vector $\boldsymbol{f} = [f_1, f_2, \dots, f_q] \in \mathbb{R}^q$ containing the frequencies
determining the frequency bands required.
for $j := 1$ to $q - 1$ do
    Create $\boldsymbol{p}_i$ with frequencies $f_A = \boldsymbol{f}_j$ and $f_B = \boldsymbol{f}_{j+1}$ as shown in (11.4).
    Create matrix $\boldsymbol{P}_j = \mathrm{diag}(\boldsymbol{p}_j)$.
    Use (11.9) to tune a model $\mathcal{M}_j$.
    Obtain $\hat{\boldsymbol{y}}_j$ through (11.10) using $\mathcal{M}_j$ and $\boldsymbol{U}_{test}$.
Estimate $\hat{\boldsymbol{y}}$ using (11.24).
```

For Partial FD-LSSVM a complexity analysis can be done in a similar way to that of FD-LSSVM.

$$\begin{aligned} EDF = \;& \sum_{j=1}^{q-1} \mathrm{tr}\left(\boldsymbol{A}\left(\boldsymbol{\Omega}_j\left(\boldsymbol{M}_j^{-1} - \boldsymbol{M}_j^{-1}\frac{\boldsymbol{J}}{c_j}\boldsymbol{M}_j^{-1}\right) + \frac{\boldsymbol{J}}{c_j}\boldsymbol{M}_j^{-1}\right)\boldsymbol{A}^\top \boldsymbol{P}_j\right) \\ & + \mathrm{tr}\left(\boldsymbol{A}\left(\boldsymbol{\Omega}_h\left(\boldsymbol{Z}_h^{-1} - \boldsymbol{Z}_h^{-1}\frac{\boldsymbol{J}}{c_h}\boldsymbol{Z}_h^{-1}\right) + \frac{\boldsymbol{J}}{c_h}\boldsymbol{Z}_h^{-1}\right)\boldsymbol{A}^\top \boldsymbol{Q}_h\right), \end{aligned} \tag{11.25}$$

with $\boldsymbol{Z}_h$ and c_h as defined in (C.3).

11.3 Results

The methods were applied to to 4 real life data examples and 1 simulation example whose data sets are publicly available[2]. In addition, the proposed methodologies were also applied to synthetic examples: two Hammerstein and two Wiener systems. Note that in order to be able to compare between the results of different examples the normalized mean absolute error (%MAE) is used as defined in Appendix D.

[2] http://http://homes.esat.kuleuven.be/∼smc/daisy/daisydata.html

11.3.1 Data set examples

A description of the data sets is presented and for each one, the division vectors used are offered in normalized frequency.

1. **Data of a CD-player arm**
   This data set corresponds to the mechanical construction of a CD player arm and consists of two inputs and two outputs. The inputs are the forces of the mechanical actuators while the outputs are related to the tracking accuracy of the arm. The data was measured in closed loop, and then through a two-step procedure converted to open loop equivalent data. The inputs are highly colored.The division vector used for the first and second outputs are $\boldsymbol{f}^{(1)} = [0, 0.14, 0.2, 0.3, 0.4, 0.5]$ and $\boldsymbol{f}^{(2)} = [0, 0.16, 0.4, 0.5]$ respectively.

2. **Wing flutter data**
   The data set shows wing flutter data and consists of one input and one output. However, due to industrial secrecy agreements no more details are revealed. Again, the input is highly colored. The division vector used is $\boldsymbol{f} = [0, 0.18, 0.32, 0.5]$.

3. **Heat flow density through a two layer wall**
   This data set depicts heat flow density through a two layer wall (i.e. brick and insulation layer) and consists of two inputs and one output. The inputs are the internal and external temperature of the wall while the output is the heat flow density through the wall. The division vector used is $\boldsymbol{f} = [0, 0.1, 0.25, 0.5]$.

4. **Data from a test setup of an industrial winding process**
   The main part of the plant is composed of a plastic web that is unwinded from first reel (unwinding reel), goes over the traction reel and is finally rewinded on the the rewinding reel. Reel 1 and 3 are coupled with a DC-motor that is controlled with input setpoint currents i_1 and i_3. The angular speed of each reel (s_1, s_2 and s_3) and the tensions in the web between reel 1 and 2 (t_1) and between reel 2 and 3 (t_3) are measured by dynamo tachometers and tension meters. s_1, s_2, s_3, i_1 and i_3 are the five inputs and t_1 and t_2 the two outputs. For the first output, the division vector used is $\boldsymbol{f}^{(1)} = [0, 0.13, 0.5]$ while for the second one it is $\boldsymbol{f}^{(2)} = [0, 0.10, 0.2, 0.38, 0.5]$.

5. **Simulation data of a pH neutralization process in a stirring tank**
   This data set represents simulation data of a pH neutralization process in a constant volume stirring tank. The volume of the tank is 1100 liters, the concentration of the acid solution (HAC) is 0.0032 Mol/l and the concentration of the base solution (NaOH) is $0, 05$ Mol/l. The data set consists of two inputs (i.e. acid solution flow in liters and base solution flow in liters) and one output (i.e. pH of the solution in the tank). The division vector used is $\boldsymbol{f} = [0, 0.1, 0.3, 0.5]$.

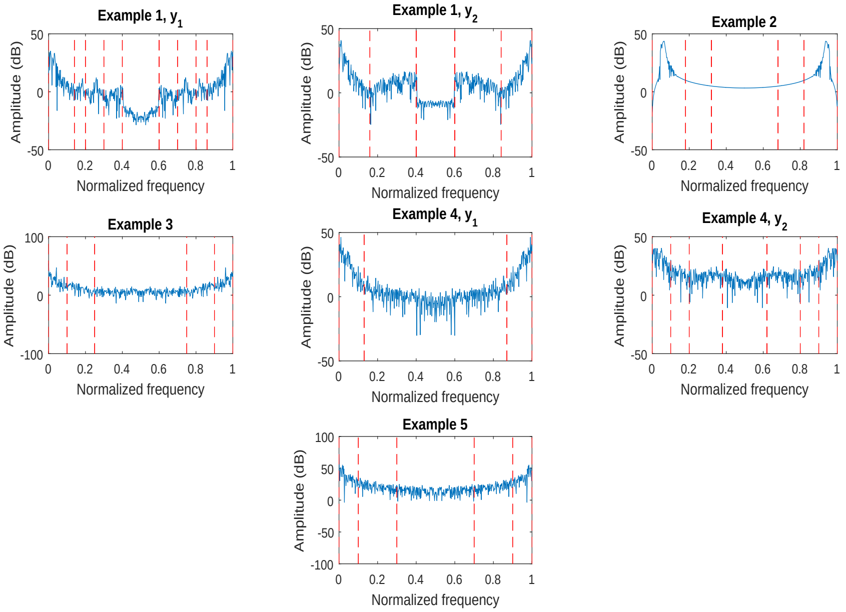

Example 1, y_1 Example 1, y_2 Example 2 Example 3 Example 4, y_1 Example 4, y_2 Example 5

Figure 11.2: Frequency domain representation of the training set outputs of the different data sets used. Division vectors are represented by the dashed lines.

In Fig. 11.2 the training set outputs of the different data sets used are presented accompanied by the frequency bands corresponding to the respective division vectors.

Method steps

In order to illustrate the procedure involved in FD-LSSVM the estimation of the first output of Example 1 is presented.

First, by observation of the output in the training data $\boldsymbol{f} \in \mathbb{R}^{p+2}$ is selected. Then for each pair $\boldsymbol{f}_i, \boldsymbol{f}_{i+1}$ for $i = 1, \ldots, p+1$ a vector $\boldsymbol{p}$ is generated and with it the corresponding diagonal matrix $\boldsymbol{P}$. Using (11.9) $p + 1$ models $\mathcal{M}_i$ are found. Finally all the resulting models are merged together as shown in (11.13).

In Figs. 11.3 to 11.7 the estimation from the resulting models are illustrated. Note

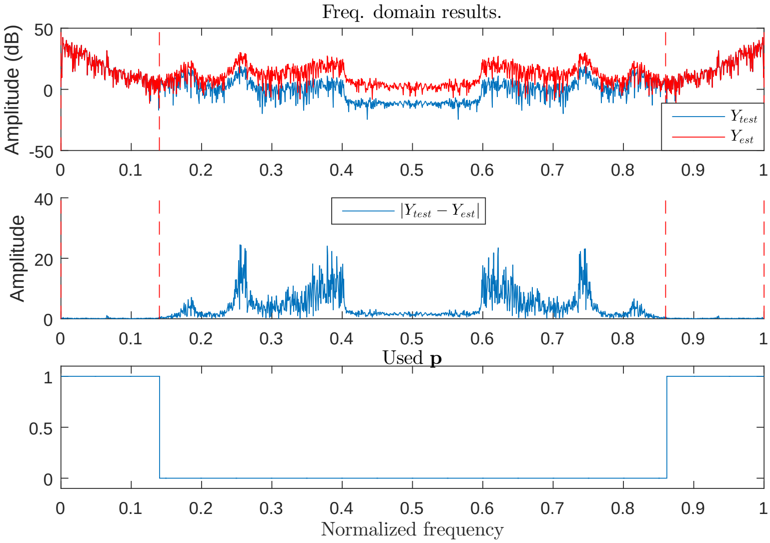


Figure 11.3: Result of applying the found model m_1 to $\boldsymbol{U}_{test}$. (Up) Overlapping in the frequency domain of $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$. (Center) Magnitude of the difference between $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$ in the frequency domain. (Bottom) $\boldsymbol{p}$ used to construct $\boldsymbol{P}$.

that for each of the models, the frequency band delimited by $\boldsymbol{p}$ has a particularly good accuracy.

It is interesting to compare the behavior of FD-LSSVM in the frequency domain to that of the standard NARX LSSVM. Fig. 11.8 illustrates the later.

Fig. 11.9 illustrates the resulting estimations allowing a comparison between FD-LSSVM, NARX LSSVM and Partial FD-LSSVM. The first 500 samples were used as training data while the rest of the set was employed as test set. For all the methods, the input and output delays were set at 5. For Partial FD-LSSVM only the first frequency band from the division vector used in FD-LSSVM was employed.

In Fig. 11.10 the estimation of FD-LSSVM for the first output of Example 1 is presented in the form of scatter plots for different division vectors $\boldsymbol{f}$ arbitrarily chosen. As can be seen, the results are even better for some of the chosen $\boldsymbol{f}$ but are in general similar. This shows on one hand that the method is robust regarding the selection of the division vector while in the other, it shows that even better results can be achieved with a careful chosing of $\boldsymbol{f}$.

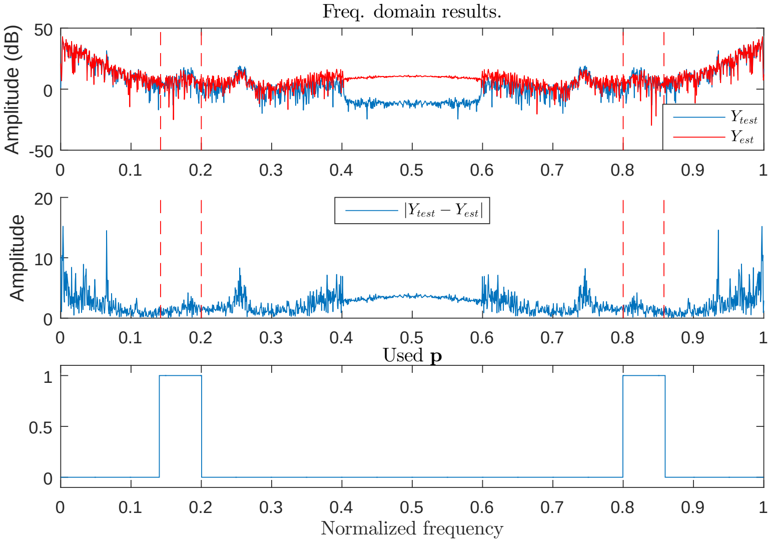


Figure 11.4: Result of applying the found model m_2 to $\boldsymbol{U}_{test}$. (Up) Overlapping in the frequency domain of $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$. (Center) Magnitude of the difference between $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$ in the frequency domain. (Bottom) $\boldsymbol{p}$ used to construct $\boldsymbol{P}$.

Results comparison

For each of the examples the first 500 samples were used as training data while the rest of the set was employed as test set. For all the methods, the input and output delays were set at 5. For the Partial FD-LSSVM method only the first frequency band was used.

In Fig. 11.11 the results for 100 Monte Carlo simulations are presented for each of the examples and for the two proposed methods and NARX LS-SVM. A summary of such results is offered in Table 11.1 where the median of each Monte Carlo simulation is shown.

As can be seen, the proposed methods represent a considerable improvement in accuracy for the presented examples. In particular FD-LSSVM shows great promise for problems involving nonlinear dynamical systems. To quantify these improvements, in Table 11.2 the $\%$MAE improvement (i.e. $100\left(1 - \frac{\%MAE_{Method}}{\%MAE_{LSSVM}}\right)$) is presented for each example and each method. For FD-LSSVM the improvement is in average 50.0261%

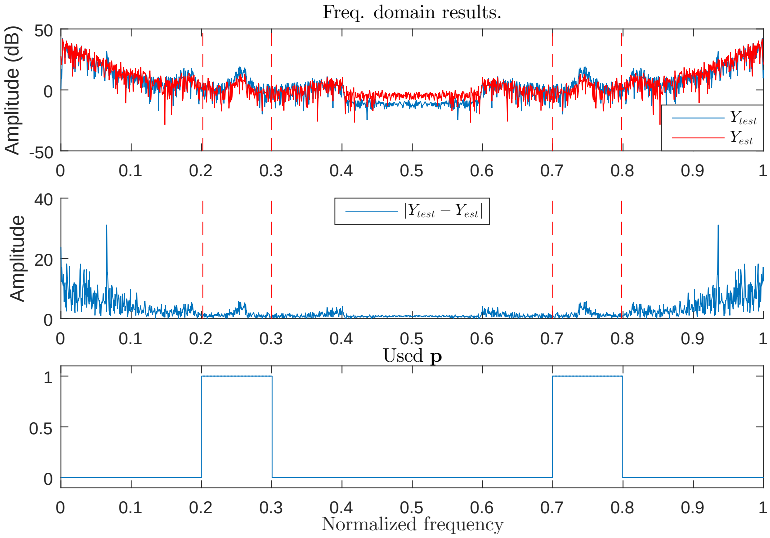


Figure 11.5: Result of applying the found model m_3 to $\boldsymbol{U}_{test}$. (Up) Overlapping in the frequency domain of $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$. (Center) Magnitude of the difference between $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$ in the frequency domain. (Bottom) $\boldsymbol{p}$ used to construct $\boldsymbol{P}$.

Table 11.1: $\%MAE$ Comparison. Median values are offered for 100 Monte Carlo simulations in the test set for each case.

	Output	FD-LSSVM	Partial FD-LSSVM	NARX LSSVM
Example 1	y_1	1.1471	3.9897	7.1057
	y_2	1.5109	1.5526	2.2523
Example 2	y	0.77256	1.3419	1.2969
Example 3	y	1.4623	2.0295	2.5253
Example 4	y_1	1.7287	2.6198	4.7319
	y_2	1.177	2.8827	3.3534
Example 5	y	3.0965	3.7028	3.9963

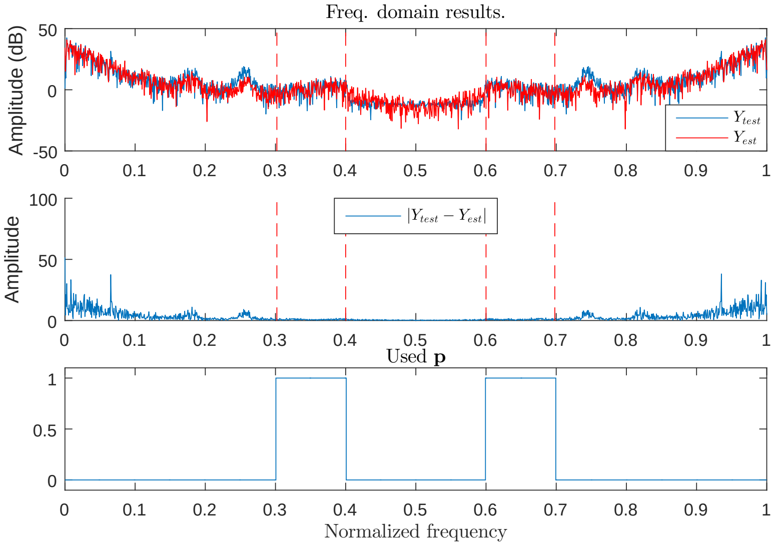


Figure 11.6: Result of applying the found model m_4 to $\boldsymbol{U}_{test}$. (Up) Overlapping in the frequency domain of $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$. (Center) Magnitude of the difference between $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$ in the frequency domain. (Bottom) $\boldsymbol{p}$ used to construct $\boldsymbol{P}$.

Table 11.2: Percentage of improvement in the $\%MAE$ wrt. LS-SVM.

	Output	FD-LSSVM	Partial FD-LSSVM
Example 1	y_1	83.8566	43.8521
	y_2	32.9175	31.0660
Example 2	y	40.4303	-3.4698
Example 3	y	42.0940	19.6333
Example 4	y_1	63.4671	44.6353
	y_2	64.9013	14.0365
Example 5	y	22.5158	7.3443

while for Partial FD-LSSVM it is 22.4425%.

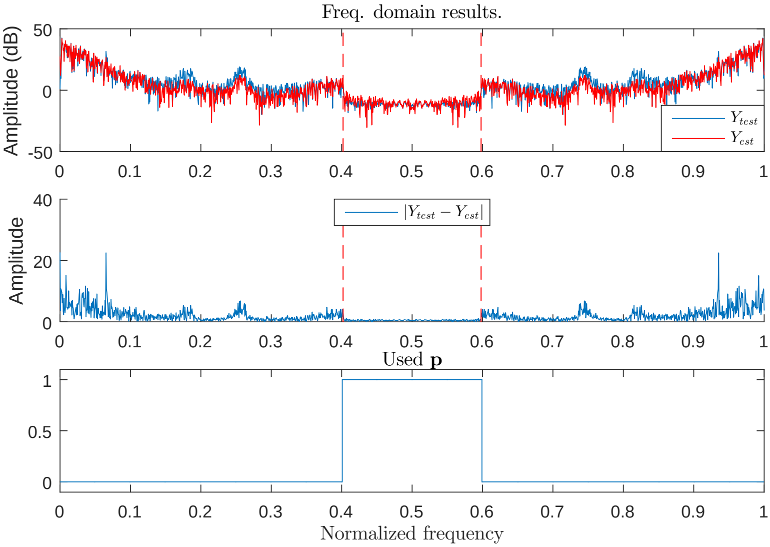


Figure 11.7: Result of applying the found model m_5 to $\boldsymbol{U}_{test}$. (Up) Overlapping in the frequency domain of $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$. (Center) Magnitude of the difference between $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$ in the frequency domain. (Bottom) $\boldsymbol{p}$ used to construct $\boldsymbol{P}$.

11.3.2 Synthetic examples

Using the LTI and nonlinear blocks presented in Fig. 11.12 two Hammerstein and two Wiener systems were created. To do so, linear block 1 and nonlinearity 1 were cascaded to create a Wiener and a Hammerstein system. Similarly, linear block 2 and nonlinearity 2 were cascaded to create a second set of Wiener and Hammerstein systems.

On each of the cases the training and test sets consisted of 500 and 1000 samples respectively. For the training set an initial signal as the one shown in Fig. 11.13 was created and then randomly permuted. In all cases, the inputs for the test set were drawn from an uniform distribution between -10 and 10. For the 4 systems tested, the division vector used was $\boldsymbol{f} = [0, 0.16, 0.33, 0.5]$ (i.e. normalized frequency). The delays used for NARX-LSSVM, FD-LSSVM and Partial FD-LSSVM are the actual delays of the underlying systems and for the Partial FD-LSSVM method only the first frequency band was used. The result of 100 Monte Carlo simulations is presented in Fig. 11.14

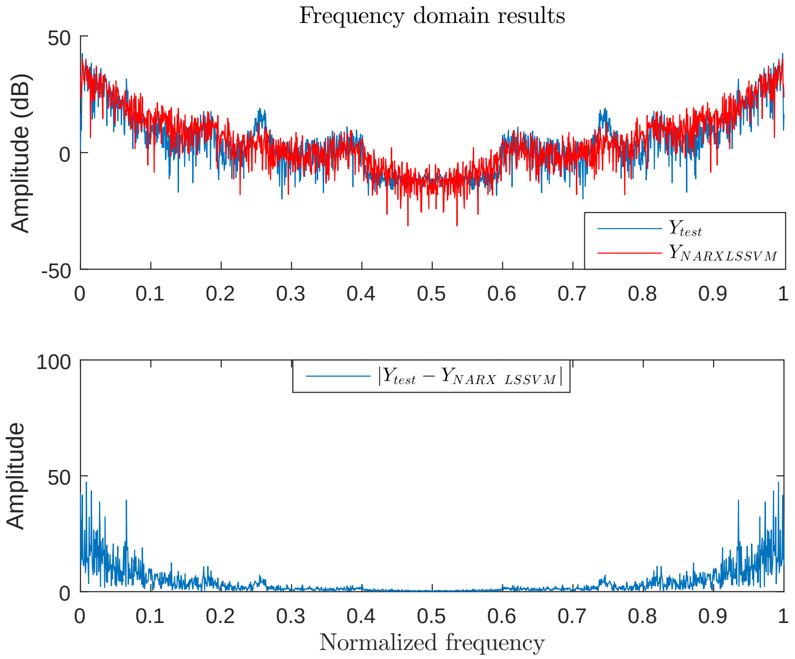


Figure 11.8: Result of applying the NARX LSSVM model to $\boldsymbol{U}_{test}$. (Up) Overlapping in the frequency domain of $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$. (Bottom) Magnitude of the difference between $\boldsymbol{y}_{test}$ and $\boldsymbol{y}_{FD-LSSVM}$ in the frequency domain.

Table 11.3: Percentage of improvement in the $\%MAE$ wrt. LS-SVM.

	FD-LSSVM	Partial FD-LSSVM
Hammerstein 1	73.8644	12.7322
Hammerstein 2	54.0519	16.7089
Wiener 1	58.5398	4.0437
Wiener 2	55.1123	10.1361

Once more, as can be seen, the proposed methods outperform standard NARX-LSSVM. In Table 11.3 a comparison is presented.

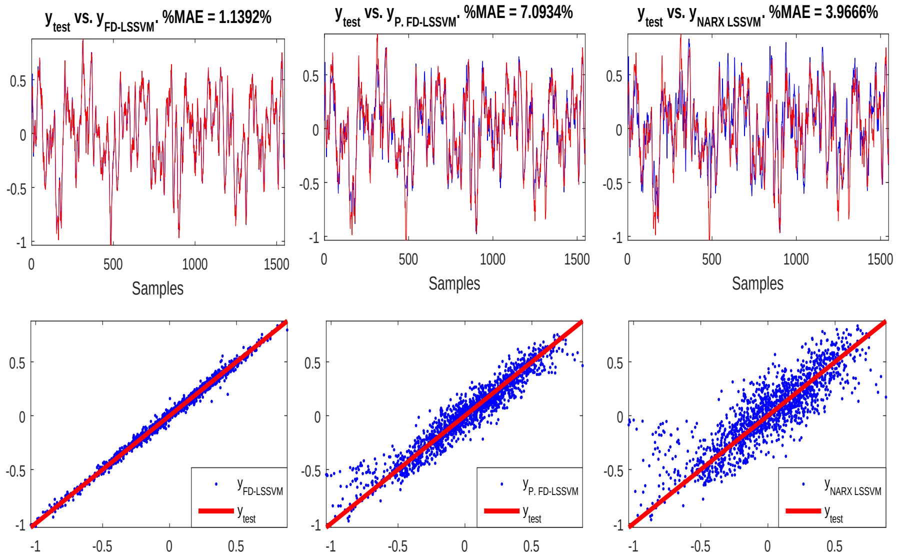


Figure 11.9: Resulting estimation of $\boldsymbol{y}_{test}$. (Up) Overlapping in the time domain of $\boldsymbol{y}_{test}$ and the different estimations $\hat{\boldsymbol{y}}$. (Bottom) Scatter plots emphasizing the accuracy of the methods. The line corresponds to a perfect fit while the dots show the evaluated method. (Left) Results for FD-LSSVM. (Center) Results for the Partial FD-LSSVM method. (Right) Results for NARX LSSVM.

11.4 Conclusions

In this chapter, two new methodologies to model problems involving dynamics have been presented, namely FD-LSSVM and Partial FD-LSSVM. Both of them were tested in several real life data sets and a simulation one and several synthetic examples and showed to significantly improve the performance of NARX LS-SVM. Given the NARX structure of FD-LSSVM and Partial FD-LSSVM, these methodologies are particularly useful for modeling nonlinear dynamical systems.

The main contribution of the presented methods lies in the fact that they create different models focusing on specific frequency bands and then merge those models together. Through this procedure, it is possible to find a resulting model that performs much better than the standard NARX-LSSVM. It was shown through the examples that the two proposed methods can handle systems with different numbers of inputs and

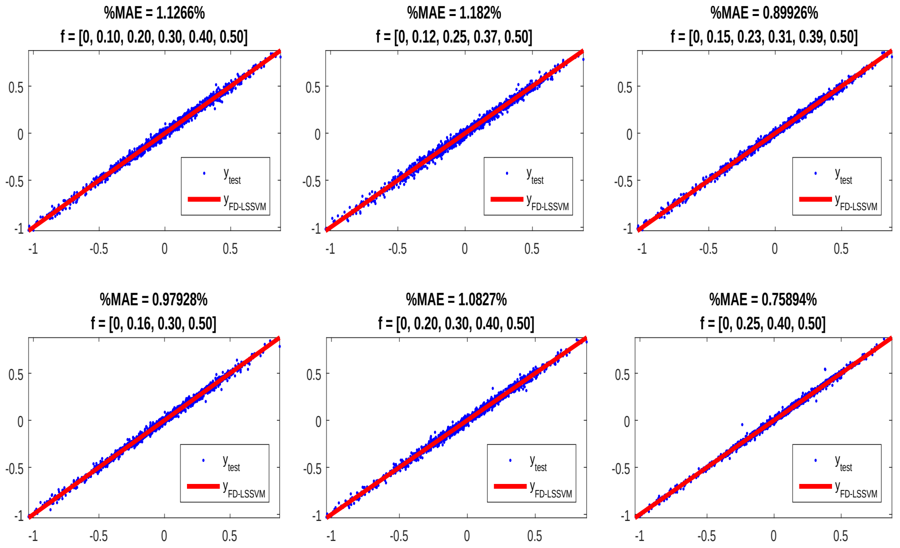


Figure 11.10: Resulting estimation of FD-LSSVM for the first output of Example 1. Scatter plots are presented for different division vectors $\boldsymbol{f}$ arbitrarily chosen. The line corresponds to a perfect fit while the dots show the FD-LSSVM estimation for the used $\boldsymbol{f}$.

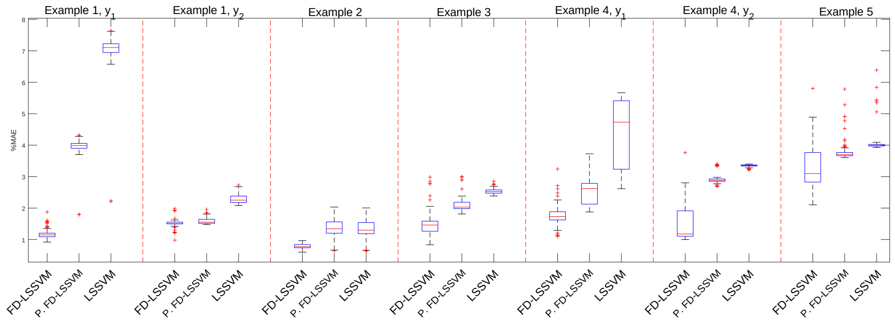


Figure 11.11: $\%MAE$ for 100 Monte Carlo simulations of the considered methods in different examples.

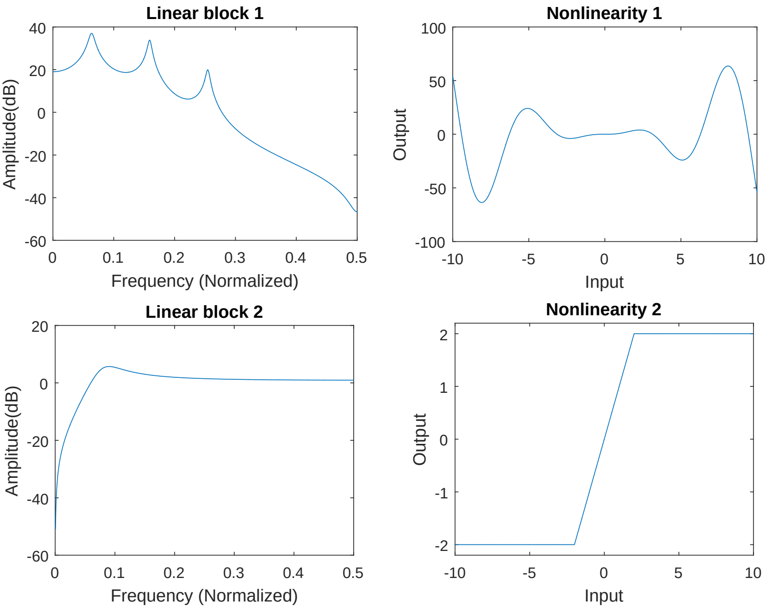


Figure 11.12: LTI and nonlinear blocks used to create the Wiener and Hammerstein systems.

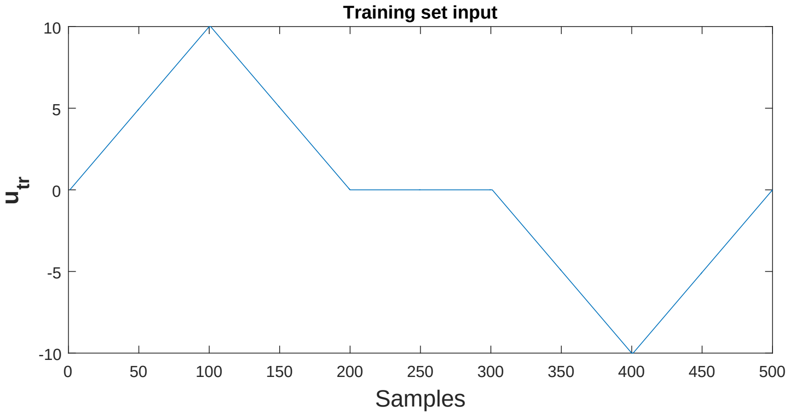


Figure 11.13: Initial signal for the training set.

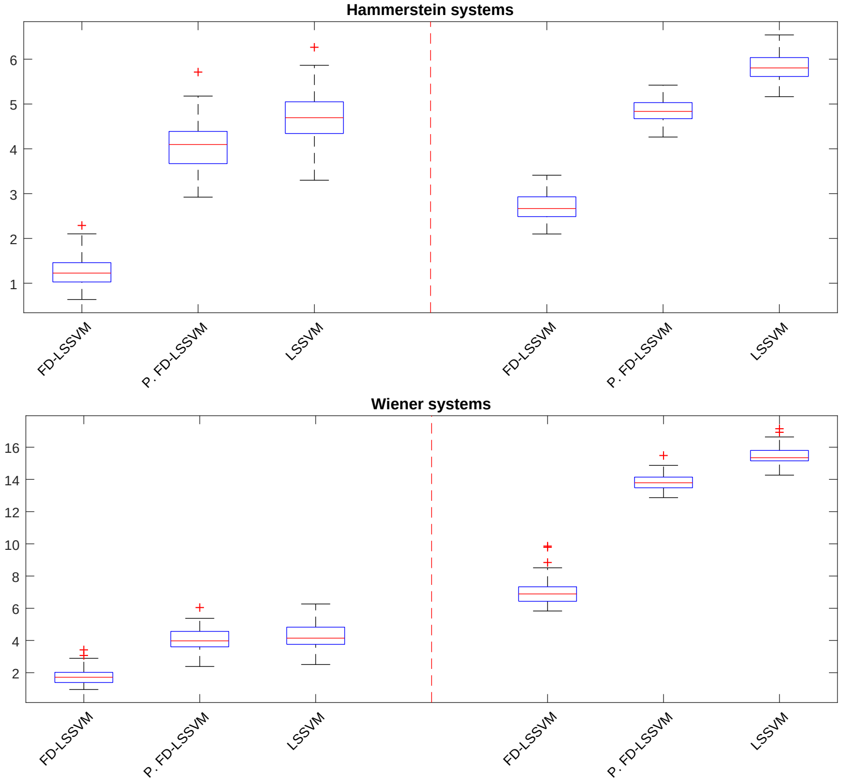


Figure 11.14: Results for 100 Monte Carlo simulation. (Top) Hammerstein systems. (Bottom) Wiener systems. In each figure the results to the left correspond to the combination of the linear block 1 and nonlinearity 1. On the right, the results for the combination of linear block 2 and nonlinearity 2 are presented.

outputs.

FD-LSSVM showed a greater accuracy; however it is a computationally demanding method as $p + 1$ models have to be estimated (for a division vector $\boldsymbol{f} \in \mathbb{R}^{p+2}$). On the other hand, Partial FD-LSSVM can still achieve very good accuracy while requiring less computation as it implies the estimation of fewer models (i.e. it requires $q + 1$ models with $q < p$). By presenting both methods we offer an alternative where a tradeoff between the processing time and the accuracy of the results is possible.

Currently, the division vectors are manually created after inspection of the outputs in the frequency domain. Nonetheless, it was shown that the method is robust to such selections although the results could further improve with a careful composition of $\boldsymbol{f}$. Future work could include the automatic determination of the division vectors. For instance, the selection could be done by partitioning the frequency spectrum in accordance to the accumulated power.

Chapter 12

Conclusions

Throughout this thesis, new techniques developed for BONL system identification were presented. These techniques imply the use of kernel methods in the form of different formulations of LS-SVM. One of the common denominators of the presented methods is the fact that knowledge about the structure of the systems is incorporated into black-box modeling schemes greatly extending their capabilities. This is illustrated by often comparing the performance of the presented methods against that of black-box modeling techniques like NARX LS-SVM. The results reported are in line with what would be expected, this is: the offered methods outperform NARX LS-SVM as they incorporate more information about the system.

The developed methods were applied in some cases to system identification benchmarks like the Wiener-Hammerstein and the Silverbox data sets (see Chapter 10) or the DAISY data set (see Chapter 11) and in others cases to simulation examples where specific input signals are designed for the particular cases. In almost every chapter, the presented methodologies were compared to different state of the art methods showing a very good comparative performance. This implies that the offered alternatives introduced in this thesis are attractive options when dealing with the identification of BONL systems. This is particularly interesting when considering that not only an increased performance is offered but also a great deal of flexibility to deal with a variety of difficult problem classes. Furthermore, some of the introduced methods were extended to include MIMO structures extending the applicability of the offered methodologies.

12.1 Summary and contributions

12.1.1 Part I

In this part the BLA is used in combination with LS-SVM for the identification of Hammerstein and Wiener systems. For all the methods presented, it is necessary for the BLA to accurately represent the LTI block as its results are subsequently integrated into an LS-SVM formulation. This is particularly true for the methods presented in Chapters 3 and 4 where the coefficients of the transfer function estimated by the BLA are used directly into the LS-SVM reformulation. The work presented in Chapter 2 is more robust against this as the introduced errors can be perceived as disturbances in the intermediate variable estimated initially and can be corrected in posterior stages.

Chapter 2

A method is presented for the identification of Hammerstein systems. Here, the BLA is used to obtain a preliminary nonparametric model of the LTI block of the underlying Hammerstein system. It is shown how it is possible to use the inversion of this block and the measured output to make an estimation of the intermediate variable even in the presence of measurement noise. Afterward, this intermediate variable is used in combination with the known input to estimate a nonlinear block using LS-SVM. Finally, the parametric LTI block is recalculated. The method was tested in three examples, two of them with hard nonlinearities, and was compared with four other methods showing very good performance in all cases.

This chapter provides two main contributions. First the method itself is shown to provide very good results when compared with other state of the art methods in difficult examples containing hard nonlinearities. Secondly, it is shown how the regularization allows to bypass the usual problems associated with the noise backpropagation when the inversion of the estimated linear block is used to compute the intermediate variable.

Chapter 3

In this chapter once more a method is presented for the identification of Hammerstein systems. In this case the BLA is used to obtain an approximation to the coefficients of the transfer function of the LTI block and these coefficients are integrated into an LS-SVM reformulation to model the system. The method was tested in two examples and compared against NARX LS-SVM. Given that there is no previous information incorporated in the NARX LS-SVM it was to be expected that the proposed method would perform better as indeed happened.

The main contribution of this chapter is that the method shows a way to introduce previous information of the system into an LS-SVM formulation.

Chapter 4

The same concept used in Chapter 3 is applied here to Wiener systems. Even though the methods are conceptually similar, the mathematical development is radically different. In here, the BLA is used once more to approximate the coefficients of the transfer function of the LTI block. Those coefficients are used for an LS-SVM reformulation that ends up becoming a standard LS-SVM problem.

This chapter has as its main contribution the method presented for Wiener system identification and the illustration of how by mixing techniques it is possible to integrate additional information about the model into a black box modeling technique.

12.1.2 Part II

In this part novel methods for system identification of block structured models are offered where the common denominator is the use of the steady state time response of the systems. The presented methods can provide very accurate models of the underlying systems. However, it could be argued that they have as a disadvantage that long times are required for the necessary measurements to take place, although clearly this depends on the specific system to be modeled and the number of samples desired. This alleged disadvantage is particularly obvious in Chapter 6 where the MIMO Hammerstein case is considered.

Chapters 5 and 6

In Chapter 5 a new method for identification of Hammerstein systems is presented. To do this, a specially designed multi-step signal is used followed by LS-SVM to identify the nonlinear block. Once a model of the nonlinear block is available, an estimation of the intermediate variable can be obtained. With this estimation and the known output the LTI block can be modeled. The work presented in Chapter 6 extends the work from Chapter 5 to the MIMO case. Once more, specially designed signals are used to straightforwardly model the nonlinear part and, once this is done, the intermediate variables can be estimated. With the estimated intermediate variable and the known outputs the linear part can be modeled directly.

These methods give a way to accurately model the nonlinearity of SISO and MIMO Hammerstein systems in a straightforward manner and are some of the more accurate methods introduced in this thesis. Beside the methods themselves, the most notable contribution of these Chapters is the fact that a way to directly model the nonlinear block of SISO and MIMO Hammerstein systems is presented.

Chapter 7

Here we propose a new methodology for identifying Wiener systems using the data acquired from two separate experiments. In the first experiment, we feed the system with a sinusoid at a prescribed frequency and use the steady state response of the system to estimate the static nonlinearity. In the second experiment, the estimated nonlinearity is used to identify a model of the linear block feeding the system with a persistently exciting input. Both: parametric and nonparametric approaches to estimate the static nonlinearity are presented.

The main contribution in this chapter, besides the method itself, is showing that for Wiener systems, a poorly exciting signal such as a sinusoid can help estimating part of the system by means of relatively simple least-squares based procedures.

12.1.3 Part III

In this part we take advantage of the structure of Hammerstein systems to extract information about the dynamics of the system. This is done by applying specific input signals to the system that allow the estimation of a rescaled version of its impulse response.

Chapters 8 and 9

In Chapter 8 a methodology for identifying SISO Hammerstein systems is presented where an impulse signal is used. The corresponding output of that signal allows the construction of an impulse response matrix that can be used into an LS-SVM reformulation leading to a model of the whole system. In Chapter 9 this approach is extended to the MIMO Hammerstein system case. Here, the estimation of the impulse response matrix is more elaborated but it is shown that the required procedure is still relatively simple as certain parameters can be arbitrarily chosen.

One of the main advantages of these methods comes from the fact that they are flexible concerning the class of systems they can model and that no previous knowledge about the underlying non-linearities is required except for very mild assumptions. The methods are shown to perform well in the presence of white Gaussian noise. Also, in the case of Chapter 9, it naturally adapts to handle different numbers of inputs and outputs. Finally, the methods incorporate information about the structure of the system but still the solution of the model follows from a linear system of equations.

The presented methods are of a more practical nature than those offered in Chapters 5 and 6 as the time required for the experiments is greatly reduced while still being very accurate.

12.1.4 Part IV

In the final part of the thesis two additional works were presented more related to the machine learning field than to that of system identification.

Chapter 10

Two methodologies were presented in this chapter which allow to make a tradeoff between the accuracy of the model and its complexity, this is: the complexity of the model could be greatly reduced in exchange for a small decrease in its accuracy. To measure the complexity, the effective degrees of freedom were used. The two techniques rely on the use of Fixed Size LS-SVM and once an approximation to the feature map is estimated, an SVD decomposition of the inner product of the input vectors in the feature space takes place followed by a truncation of the less relevant singular values. The resulting matrix is used instead of the originally decomposed matrix. Both techniques were tested in the Wiener-Hammerstein and the Silverbox data sets.

In this chapter the SVD truncation of FS-OLS and FS-RR methodologies are contributed and are shown to very efficiently reduce the effective degrees of freedom of Fixed-Size kernel models under an SVD truncation scheme without much loss of generalization performance.

Chapter 11

Given the NARX structure of FD-LSSVM and Partial FD-LSSVM, these methodologies are particularly useful for modeling nonlinear systems. In both methods, a reformulated version of LS-SVM is used were different models are created. For each of these models a specific band in the frequency spectrum is emphasized and the models are finally merged together to create the final model of the system. Both methodologies were tested in several real life data sets and a simulation one as well as several synthetic examples. It was shown through the examples that the proposed methods can handle systems with different numbers of inputs and outputs. Also, both methods showed a significantly improved performance compared to the standard NARX LS-SVM.

The main contribution of the presented methods lies in the fact that they create different models focusing on specific frequency bands which are merged together afterward. Throughout this procedure, it is possible to find a resulting model that performs much better than the standard NARX LS-SVM.

Currently, the frequency division vectors are manually created after inspection of the outputs in the frequency domain. Although the method is robust to such selections, the results could further improve with a more careful composition of the frequency division vector.

12.2 Future work

In BONL system identification, kernel methods have been successfully applied in the past for certain classes of model structures. This thesis presents contributions in this area, which is at the interface between nonlinear system identification and machine learning, by combining and integrating the best of both paradigms and employing parametric and kernel-based approaches. However, there is still a lot of research that can be done. We will state some possible future research topics next:

- The presented methods have shown to be very promising when dealing with Hammerstein or Wiener systems. Extending these methods to include more complex systems like Wiener-Hammerstein or Hammerstein-Wiener would widen their application area and improve the current state of the art in BONL system identification.
- The methods presented in Part I rely on the use of the BLA, however other methods could be used to estimate the linear blocks. With this in mind, the method from Chapter 2 could be for instance extended to the MIMO case when used in combination with that of Chapter 9.
- In Chapters 5 and 6 although the methods are very accurate, the initial input signal is not completely used. Instead, only the last samples of every step are taken into account. A good improvement could be to make use of those unused samples for the estimation of the linear part.
- Future work for the FD-LSSVM and Partial FD-LSSVM approaches should include the automatic determination of the frequency division vectors used. For instance, the selection could be done by partitioning the frequency spectrum in accordance with the accumulated power. Another option is to consider the selected frequencies of the division vectors as additional tuning parameters.

Appendix A

Best Linear Approximation

The best linear approximation (BLA) of a PISPO (i.e. Period In Same Period Out) system[1] with input $u(t)$ and output $y(t)$ is defined as the linear system whose output approximates the system's output best in mean-square sense, i.e.

$$G_{BLA}(k) := \arg\min_{G(k)} E_u \left\{ \|\tilde{Y}(k) - G(k)\tilde{U}(k)\|_2^2 \right\} , \tag{A.1}$$

with

$$\begin{cases} \tilde{u}(t) = u(t) - E\{u(t)\} \\ \tilde{y}(t) = y(t) - E\{y(t)\}, \end{cases}$$

where G_{BLA} is the frequency response function (FRF) of the BLA, and where the expectation in (A.1) is taken with respect to the random input $u(t)$.

It is assumed that the mean values are removed from the signals when a BLA is calculated. The notations $u(t)$ and $y(t)$ will be used instead of $\tilde{u}(t)$ and $\tilde{y}(t)$.

If the BLA exists, the minimizer in (A.1) can be found as

$$G_{BLA}(k) = \frac{S_{YU}(k)}{S_{UU}(k)} , \tag{A.2}$$

where the expectation in the cross-power and auto-power spectra is again taken with respect to the random input $u(t)$.

[1] A PISPO system is a system that when excited with a periodic signal $u(t)$ produces a periodic output $y(t)$ (in steady state) with the same period length as $u(t)$. Hammerstein systems are included in the class of PISPO systems.

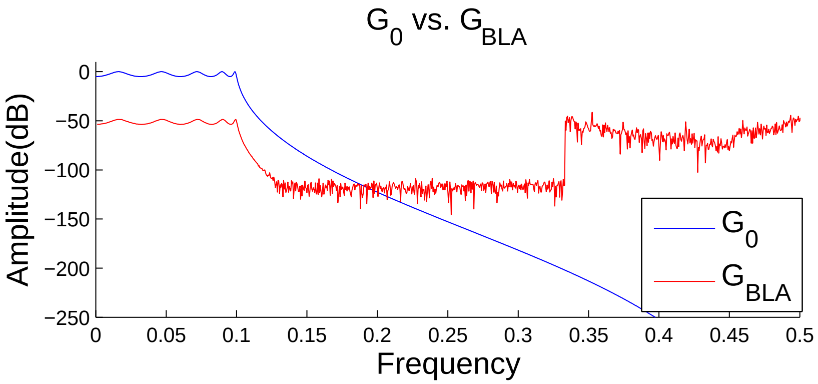


Figure A.1: Example: comparison of the magnitudes of $G_{BLA}(k)$ and the actual transfer function $G_0(k)$. $G_0(k)$ corresponds to a 10th order Chebyshev lowpass digital filter with normalized passband edge frequency 0.2 and 5 dB of peak-to-peak ripple in the passband.

Note that for periodic excitations with a fixed amplitude spectrum $|\boldsymbol{U}(k)|$ (such that $E_u\left\{|\boldsymbol{U}(k)|^2\right\} = |\boldsymbol{U}(k)|^2$), (A.2) reduces to (J. Schoukens, Pintelon, & Rolain, 2012):

$$G_{BLA}(k) = E_u\left\{\frac{Y(k)}{U(k)}\right\} . \tag{A.3}$$

In this work random-phase multisine excitations (Pintelon & Schoukens, 2012) are used for calculating the BLA. These asymptotically Gaussian distributed signals are periodic and the BLA can thus be estimated by averaging (A.3) over a number of phase realizations of the multisine. This is the main idea in the robust method (J. Schoukens et al., 2012), which provides nonparametric estimates of the BLA, the noise variance, the nonlinear variance, and the total (i.e. noise plus nonlinear) variance. Alternatively, the fast method (Pintelon & Schoukens, 2012) can be used to calculate the BLA from only one phase realization, although some restrictions apply (J. Schoukens et al., 2012).

For Gaussian distributed inputs $u(t)$, it follows from Bussgang's theorem (Bussgang, 1952) that the BLA of a Hammerstein system is proportional to the underlying linear dynamic system.

An example of a resulting $G_{BLA}(k)$ compared to the actual transfer function $G_0(k)$ is shown in Fig. A.1. As it can be seen, $G_{BLA}(k)$ resembles quite well the shape of $G_0(k)$ up to a certain frequency and up to a certain scaling factor.

It is important to highlight that the reliability of $G_{BLA}(k)$, the model found, decreases as the frequency grows apart from the band of interest. This can be seen for instance in the fluctuations present in Fig. A.1 from 12% of the sampling frequency (i.e. the signal-to-noise ratio is smaller around these frequencies than for lower ones). Note that the sudden perturbation at 33% of the sampling frequency is due to the lack of excitation of the following frequencies and thus is nothing more than an artifact due to the chosen excitation signal.

On top of the nonparametric estimate, a parametric transfer function model could be estimated using a weighted least-squares estimator (J. Schoukens, Dobrowiecki, & Pintelon, 1998)

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J_N(\boldsymbol{\theta})\,, \tag{A.4a}$$

where the cost function $J_N(\boldsymbol{\theta})$ is

$$J_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{k=1}^{N} W(k)\left|\hat{G}_{BLA}(k) - G_M(k,\boldsymbol{\theta})\right|^2 . \tag{A.4b}$$

Here, $W(k) \in \mathbb{R}^+$ is a deterministic, $\boldsymbol{\theta}$-independent weighting sequence, $\hat{G}_{BLA}(k)$ is an approximation to the actual $G_{BLA}(k)$ as it is limited to a finite number of realizations of $U(k)$ and $Y(k)$, and $G_M(k,\boldsymbol{\theta})$ is a parametric transfer function model

$$\begin{aligned} G_M(k,\boldsymbol{\theta}) &= \frac{\sum_{l=0}^{n_b} b_l \exp\left(-j2\pi\frac{k}{N}l\right)}{\sum_{l=0}^{n_a} a_l \exp\left(-j2\pi\frac{k}{N}l\right)} \\ &= \frac{B_\theta(k)}{A_\theta(k)}\,, \qquad \text{(A.4c)} \\ \boldsymbol{\theta} &= \begin{bmatrix} a_0 & \cdots & a_{n_a} & b_0 & \cdots & b_{n_b} \end{bmatrix}^T , \end{aligned}$$

with the constraints $||\boldsymbol{\theta}||_2 = 1$ and the first non-zero element of $\boldsymbol{\theta}$ positive to obtain a unique parametrization.

Appendix B

Fixed Size LS-SVM

Usually, the feature map is not explicitly known when solving in the dual. This is the case for the RBF kernel for which the feature map is infinite dimensional V. N. Vapnik (1998). In order to be able to work in the primal space, it is required that either the feature map φ is explicitly known and it is finite dimensional (e.g. linear kernel case) or an approximation to φ is acquired. This can be achieved through an eigenvalue decomposition of the kernel matrix $\boldsymbol{\Omega}$ with entries $k(\boldsymbol{x}_k, \boldsymbol{x}_l)$. Given the integral equation $\int k(\boldsymbol{x}, \boldsymbol{x}_j)\phi_i(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \lambda_i \phi_i(\boldsymbol{x}_j)$ with λ_i and ϕ_i the eigenvalues and eigenfunctions related to the kernel function respectively for a variable $\boldsymbol{x}$ with probability distribution $p(\boldsymbol{x})$. An expression for a finite sized approximation of the feature map can be written then (De Brabanter et al., 2009; Espinoza et al., 2004; Espinoza, Suykens, & De Moor, 2005b):

$$\hat{\varphi}(\boldsymbol{x}) = \left[\sqrt{\lambda_1}\phi_1(\boldsymbol{x}), \sqrt{\lambda_2}\phi_2(\boldsymbol{x}), ..., \sqrt{\lambda_{n_h}}\phi_{n_h}(\boldsymbol{x})\right]^\top . \tag{B.1}$$

Through the Nyström method (Nyström, 1930; Williams & Seeger, 2000), an approximation to the integral equation is obtained by means of the sample average determining an approximation to ϕ_i leading to

$$\frac{1}{N}\sum_{k=1}^{M} k(\boldsymbol{x}_k, \boldsymbol{x}_j)\boldsymbol{u}_{ik} = \lambda_i^{(s)}\boldsymbol{u}_{ij} \tag{B.2}$$

where $\lambda_i^{(s)}$ and $\boldsymbol{u}_i$ are the sample eigenvalues and eigenvectors respectively.

A finite dimensional approximation $\hat{\boldsymbol{\varphi}}_i(\boldsymbol{x})$ can be computed for any point $\boldsymbol{x}^{(v)}$ through

$$\begin{aligned} \hat{\varphi}_i(\boldsymbol{x}^{(v)}) &= \frac{1}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^{M} \boldsymbol{u}_{ki} k(\boldsymbol{x}_k, \boldsymbol{x}^{(v)}) \\ \text{with } i &= 1, \ldots, M. \end{aligned} \tag{B.3}$$

This approximation can then be used in the primal to estimate $\boldsymbol{w}$ and b.

For large scale problems, a subsample of M datapoints (with $M \ll N$) could be selected to compute $\hat{\boldsymbol{\varphi}}$ together with estimation in the primal. This is known as Fixed-Size Least Squares Support Vector Machines (FS-LSSVM) Suykens and Vandewalle (1999). Criteria as entropy maximization has been used to select appropriate M datapoints instead of a merely random approach. For example Rényi's entropy H_R is used Girolami (2002) as:

$$H_R = -\log \int p(\boldsymbol{x})^2 d\boldsymbol{x}. \tag{B.4}$$

The higher the entropy found in the subset of M points used, the better this subset will represent the whole data set.

Once the support vectors are selected through Rényi's entropy, the problem in the primal can be represented as

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2} \sum_{i=1}^{M} (y_i - \boldsymbol{w}^\top \hat{\varphi}(\boldsymbol{x}_i) - b)^2 \tag{B.5}$$

from where the optimal $\boldsymbol{w}$ and b can be extracted directly. Note that given the selection of $M \ll N$, this is a sparse kernel model.

Appendix C

Effective Degrees of Freedom for LS-SVM

When analyzing the complexity of LS-SVM, the plain number of model parameters is not a good indicator of the complexity of the model found. This type of measurements is not suitable for techniques using regularization such as in the LS-SVM case. Instead, in this thesis the Effective Degrees of Freedom (EDF) will be used. The EDF can be calculated as the trace of the hat matrix $\boldsymbol{H}$ (also known as the smoother matrix) (Espinoza, Suykens, & De Moor, 2005a; Mallows, 1973; Spiegelhalter et al., 2002) which is defined from the expression $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$. For further insight about the effective degrees of freedom see Bishop (1995); MacKay (1992); Moody (1991).

For LS-SVM, $\boldsymbol{H}$ is calculated as follows. From (1.32)

$$\begin{array}{rcl} \boldsymbol{y}_{tr} & = & \boldsymbol{\Omega}\boldsymbol{\alpha} + \mathbf{1}_N b + \frac{\boldsymbol{\alpha}}{\gamma}, \\ \boldsymbol{y}_{tr} & = & (\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I})\boldsymbol{\alpha} + \mathbf{1}_N b. \end{array} \tag{C.1}$$

From (C.1), $\boldsymbol{\alpha}$ and b can be written as

$$\left\{ \begin{array}{rcl} \boldsymbol{\alpha} & = & (\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I})^{-1}(\boldsymbol{y}_{tr} - \mathbf{1}_N b), \\ b & = & \dfrac{\mathbf{1}_N^\top(\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I})^{-1}\boldsymbol{y}_{tr}}{\mathbf{1}_N^\top(\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I})^{-1}\mathbf{1}_N}. \end{array} \right. \tag{C.2}$$

Let us define now

$$\left\{ \begin{array}{l} c = \mathbf{1}_N^\top(\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I})^{-1}\mathbf{1}_N, \\ \boldsymbol{Z} = (\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}), \\ \boldsymbol{J} = \mathbf{1}_N\mathbf{1}_N^\top. \end{array} \right. \tag{C.3}$$

From (1.33), for the training set we have

$$\hat{\boldsymbol{y}}_{tr} = \boldsymbol{\Omega}\boldsymbol{\alpha} + \mathbf{1}_N b. \tag{C.4}$$

Therefore,

$$\begin{aligned}
\hat{\boldsymbol{y}}_{tr} &= \boldsymbol{\Omega}\boldsymbol{Z}^{-1}(\boldsymbol{y}_{tr} - \mathbf{1}_N \frac{\mathbf{1}_N^\top \boldsymbol{Z}^{-1}\boldsymbol{y}_{tr}}{c}) + \mathbf{1}_N \frac{\mathbf{1}_N^\top \boldsymbol{Z}^{-1}\boldsymbol{y}_{tr}}{c} \\
\hat{\boldsymbol{y}}_{tr} &= \boldsymbol{\Omega}\boldsymbol{Z}^{-1}\left(\boldsymbol{y}_{tr} - \frac{\boldsymbol{J}}{c}\boldsymbol{Z}^{-1}\boldsymbol{y}_{tr}\right) + \frac{\boldsymbol{J}}{c}\boldsymbol{Z}^{-1}\boldsymbol{y}_{tr} \\
\hat{\boldsymbol{y}}_{tr} &= \left(\boldsymbol{\Omega}\left(\boldsymbol{Z}^{-1} - \boldsymbol{Z}^{-1}\frac{\boldsymbol{J}}{c}\boldsymbol{Z}^{-1}\right) + \frac{\boldsymbol{J}}{c}\boldsymbol{Z}^{-1}\right)\boldsymbol{y}_{tr} \\
\hat{\boldsymbol{y}}_{tr} &= \boldsymbol{H}\boldsymbol{y}_{tr}.
\end{aligned} \tag{C.5}$$

Finally we obtain the EDF for LS-SVM as $EDF_{LS-SVM} = \text{tr}(\boldsymbol{H})$.

Appendix D

Normalized Mean Absolute error

In order to be able to compare between the results of different examples, let us have the Normalized MAE defined as shown in (D.1) for a signal with N measurements. Note that the Normalized MAE uses the noise free signal $y_{test}(t)$ and its estimated counterpart $\hat{y}_{test}(t)$.

$$\%\mathrm{MAE} = \frac{100}{N} \frac{\sum_{i=1}^{N} |\boldsymbol{y}_{test,i} - \hat{\boldsymbol{y}}_{test,i}|}{|\max(\boldsymbol{y}_{test}) - \min(\boldsymbol{y}_{test})|}. \tag{D.1}$$

This way of measuring the performance of the methods will be used often throughout this thesis.

Bibliography

Al-Duwaish, H., & Karim, M. N. (1997). A new method for the identification of Hammerstein model. *Automatica*, *33*(10), 1871–1875.

Aljamaan, I., Westwick, D. T., & Foley, M. (2014). Non-Iterative Identification of IIR Wiener Systems Using Orthogonal Polynomial. *IFAC Proceedings Volumes*, *47*(3), 487–492.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, *68*(3), 337–404.

Bai, E. W. (1998). An optimal two stage identification algorithm for Hammerstein-Wiener nonlinear systems. *Automatica*, *34*, 333–338.

Bai, E.-W. (2002). A blind approach to the Hammerstein–Wiener model identification. *Automatica*, *38*(6), 967–979.

Bai, E.-W. (2004). Decoupling the linear and nonlinear parts in Hammerstein model identification. *Automatica*, *40*(4), 671–676.

Bai, E. W., & Fu, M. (2002). A blind approach to Hammerstein model identification. *IEEE Transactions on Signal Processing*, *50*(7), 1610–1619.

Bai, E. W., & Li, K. (2010). Convergence of the iterative algorithm for a general Hammerstein system identification. *Automatica*, *46*(11), 1891–1896.

Balestrino, A., Landi, A., Ould-Zmirli, M., & Sani, L. (2001). Automatic nonlinear auto-tuning method for Hammerstein modeling of electrical drives. *IEEE Transactions on Industrial Electronics*, *48*(3), 645–655.

Bergmann, S. (1922). Über die entwicklung der harmonischen funktionen der ebene und des raumes nach orthogonalfunktionen. *Mathematische Annalen*, *86*(3), 238–271.

Beyer, W. H. (1978). CRC standard mathematical tables. *West Palm Beach, Fl.: Chemical Rubber Co., 1978, 25th ed., edited by Beyer, William H.*.

Billings, S., & Fakhouri, S. (1977). Identification of nonlinear systems using the Wiener model. *Electronics letters*, *13*(17), 502–504.

Billings, S., & Fakhouri, S. (1982). Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, *18*(1), 15–26.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford university press.

Bottegal, G., Castro-Garcia, R., & Suykens, J. A. K. (2017a). On the identification of Wiener systems with polynomial nonlinearity. In *Proceedings of the 56th IEEE Conference on Decision and Control - CDC2017 (Accepted for publication).*

Bottegal, G., Castro-Garcia, R., & Suykens, J. A. K. (2017b). A two-experiment approach to Wiener system identification. In *Internal report 17-38, ESAT-SISTA, KU Leuven (Leuven, Belgium).*

Boyd, S., & Chua, L. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on circuits and systems*, *32*(11), 1150–1161.

Boyd, S., & Chua, L. O. (1983). Uniqueness of a basic nonlinear structure. *IEEE Transactions on Circuits and Systems*, *30*(9), 648–651.

Bruls, J., Chou, C., Haverkamp, B., & Verhaegen, M. (1999). Linear and non-linear system identification using separable least-squares. *European Journal of Control*, *5*(1), 116–128.

Bussgang, J. (1952). *Crosscorrelation functions of amplitude-distorted Gaussian signals.* MIT, Cambridge, MA, USA, Tech. Rep.

Castro, R., Mehrkanoon, S., Marconato, A., Schoukens, J., & Suykens, J. A. K. (2014). SVD truncation schemes for fixed-size kernel models. In *Neural Networks (IJCNN), 2014 International Joint Conference on* (pp. 3922–3929).

Castro-Garcia, R., Agudelo, O. M., & Suykens, J. A. K. (2017a). Impulse response constrained LS-SVM modeling for MIMO Hammerstein system identification. In *Accepted for publication in the International Journal of Control.* doi: 10.1080/00207179.2017.1373862

Castro-Garcia, R., Agudelo, O. M., & Suykens, J. A. K. (2017b). Impulse Response Constrained LS-SVM modeling for Hammerstein System Identification. In *Proceedings of the 20th world congress of the international federation of automatic control* (pp. 14611–14616).

Castro-Garcia, R., Agudelo, O. M., & Suykens, J. A. K. (2017c). MIMO Hammerstein system identification using LS-SVM and steady state time response. In *Internal Report 17-23, ESAT-SISTA, KU Leuven (Leuven, Belgium).*

Castro-Garcia, R., Agudelo, O. M., Tiels, K., & Suykens, J. A. K. (2016). Hammerstein system identification using LS-SVM and steady state time response. In *Proc. of the 15th European Control Conference* (pp. 1063–1068).

Castro-Garcia, R., & Suykens, J. A. K. (2016). Wiener System Identification using Best Linear Approximation within the LS-SVM framework. In *Proc. of the 3rd Latin American Conference on Computational Intelligence.* doi: 10.1109/LA-CCI.2016.7885698

Castro-Garcia, R., Tiels, K., Agudelo, O. M., & Suykens, J. A. K. (2017). Hammerstein system identification through BLA inversion and LS-SVM techniques. *International Journal of Control.* doi: 10.1080/00207179.2017.1329550

Castro-Garcia, R., Tiels, K., Schoukens, J., & Suykens, J. A. K. (2015). Incorporating

Best Linear Approximation within LS-SVM-Based Hammerstein system identification. In *Proceedings of the 54th IEEE conference on decision and control (CDC 2015)* (pp. 7392–7397).

Castro-Garcia, R., Tiels, K., & Suykens, J. A. K. (2017). Frequency Division Least Squares Support Vector Machines. In *Internal report 17-24, ESAT-SISTA, KU Leuven (Leuven, Belgium).*

Chang, F. H. I., & Luus, R. (1971). A noniterative method for identification using the hammerstein model. *IEEE Transactions on Automatic Control*, *16*, 464—468.

Chen, S., & Billings, S. (1989). Representations of non-linear systems: the NARMAX model. *International Journal of Control*, *49*(3), 1013–1032.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Crama, P., & Schoukens, J. (2001a). First estimates of Wiener and Hammerstein systems using multisine excitation. In *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE* (Vol. 2, pp. 1365–1369).

Crama, P., & Schoukens, J. (2001b). Initial estimates of Wiener and Hammerstein systems using multisine excitation. *IEEE transactions on Instrumentation and Measurement*, *50*(6), 1791–1795.

Crama, P., & Schoukens, J. (2004). Hammerstein–Wiener system estimator initialization. *Automatica*, *40*(9), 1543–1550.

De Brabanter, K., De Brabanter, J., Suykens, J. A. K., & De Moor, B. (2011a). Approximate confidence and prediction intervals for least squares support vector regression. *IEEE Transactions on Neural Networks*, *22*(1), 110–120.

De Brabanter, K., De Brabanter, J., Suykens, J. A. K., & De Moor, B. (2011b). Kernel regression in the presence of correlated errors. *Journal of Machine Learning Research*, *12*, 1955–1976.

De Brabanter, K., Dreesen, P., Karsmakers, P., Pelckmans, K., De Brabanter, J., Suykens, J. A. K., & De Moor, B. (2009). Fixed-size LS-SVM applied to the Wiener-Hammerstein benchmark. In *Proceedings of the 15th IFAC symposium on system identification (SYSID 2009)* (pp. 826–831).

De Moor, B., De Gersem, P., De Schutter, B., & Favoreel, W. (1997). DAISY: A database for identification of systems. *Journal A*, *38*(3), 4–5.

Dempsey, E. J., & Westwick, D. T. (2004). Identification of Hammerstein models with cubic spline nonlinearities. *IEEE Transactions on Biomedical Engineering*, *51*(2), 237–245.

Eskinat, E., Johnson, S. H., & Luyben, W. L. (1991). Use of Hammerstein models in identification of nonlinear systems. *AIChE Journal*, *37*(2), 255–268.

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

Espinoza, M., Suykens, J. A. K., & De Moor, B. (2005a). Kernel based partially linear models and nonlinear identification. *IEEE Transactions on Automatic Control*, *50*(10), 1602–1606.

Espinoza, M., Suykens, J. A. K., & De Moor, B. (2005b). Load forecasting using fixed-size least squares support vector machines. In *International Work-Conference on Artificial Neural Networks* (pp. 1018–1026).

Falck, T., Dreesen, P., De Brabanter, K., Pelckmans, K., De Moor, B., & Suykens, J. A. K. (2012). Least-Squares Support Vector Machines for the identification of Wiener-Hammerstein systems. *Control Engineering Practice*, *20*(11), 1165–1174.

Falck, T., Pelckmans, K., Suykens, J. A. K., & De Moor, B. (2009). Identification of Wiener-Hammerstein systems using LS-SVMs. In *Proceedings of the 15th IFAC symposium on system identification (SYSID 2009)* (pp. 820–825).

Falck, T., Suykens, J. A. K., Schoukens, J., & De Moor, B. (2010). Nuclear norm regularization for overparametrized Hammerstein systems. In *Proceedings of the 49th IEEE conference on decision and control (CDC 2010)* (pp. 7202–7207).

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics Springer, Berlin.

Fruzzetti, K., Palazoglu, A., & McDonald, K. (1997). Nolinear model predictive control using Hammerstein models. *Journal of process control*, *7*(1), 31–41.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural networks*, *2*(3), 183–192.

Giri, F., & Bai, E.-W. (Eds.). (2010). *Block-oriented nonlinear system identification* (Vol. 1). Springer.

Giri, F., Radouane, A., Brouri, A., & Chaoui, F.-Z. (2014). Combined frequency-prediction error identification approach for Wiener systems with backlash and backlash-inverse operators. *Automatica*, *50*(3), 768–783.

Giri, F., Rochdi, Y., Brouri, A., & Chaoui, F. Z. (2011). Parameter identification of Hammerstein systems containing backlash operators with arbitrary-shape parametric borders. *Automatica*, *47*(8), 1827–1833.

Girolami, M. (2002). Orthogonal series density estimation and the kernel eigenvalue problem. *Neural computation*, *14*(3), 669–688.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural computation*, *10*(6), 1455–1480.

Goethals, I., Pelckmans, K., Suykens, J. A. K., & De Moor, B. (2005). Identification of MIMO Hammerstein models using Least-Squares Support Vector Machines. *Automatica*, *41*(7), 1263–1272.

Gomez, J. C., & Baeyens, E. (2004). Identification of block-oriented nonlinear systems using orthonormal bases. *Journal of Process Control*, *14*(6), 685–697.

Greblicki, W. (1992). Nonparametric identification of Wiener systems. *IEEE Transactions on information theory*, *38*(5), 1487–1493.

Greblicki, W. (1997). Nonparametric approach to Wiener system identification. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *44*(6), 538–545.

Greblicki, W. (2001). Recursive identification of Wiener systems. *Applied Mathematics and computer science*, *11*(4), 977–992.

Hagenblad, A., Ljung, L., & Wills, A. (2008). Maximum likelihood identification of Wiener models. *Automatica*, *44*(11), 2697–2705.

Hammerstein, A. (1930). Nichtlineare Integralgleichungen nebst Anwendungen. *Acta Mathematica*, *54*, 117–176.

Hansen, P. C. (1990). Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM Journal on Scientific and Statistical Computing*, *11*(3), 503–518.

Hasiewicz, Z., Mzyk, G., Śliwiński, P., & Wachel, P. (2012). Mixed parametric-nonparametric identification of Hammerstein and Wiener systems-a survey. *IFAC Proceedings Volumes*, *45*(16), 464–469.

Hunt, K. J., Munih, M., Donaldson, N. d. N., & Barr, F. M. (1998). Investigation of the Hammerstein hypothesis in the modeling of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, *45*(8), 998–1009.

Hunter, I., & Korenberg, M. (1986). The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biological cybernetics*, *55*(2-3), 135–144.

Ikhouane, F., & Giri, F. (2014). A unified approach for the parametric identification of SISO/MIMO Wiener and Hammerstein systems. *Journal of the Franklin Institute*, *351*(3), 1717–1727.

Janczak, A. (2007). Instrumental variables approach to identification of a class of MIMO Wiener systems. *Nonlinear Dynamics*, *48*(3), 275–284.

Jeng, J.-C., & Huang, H.-P. (2008). Nonparametric identification for control of MIMO Hammerstein systems. *Industrial & Engineering Chemistry Research*, *47*(17), 6640–6647.

Jiang, L., Liu, F., & He, Y. (2012). A non-destructive distinctive method for discrimination of automobile lubricant variety by visible and short-wave infrared spectroscopy. *Sensors*, *12*, 3498–3511.

Jurado, F. (2006). A method for the identification of solid oxide fuel cells using a Hammerstein model. *Journal of Power Sources*, *154*(1), 145–152.

Kalafatis, A. D., Wang, L., & Cluett, W. R. (2005). Identification of time-varying ph processes using sinusoidal signals. *Automatica*, *41*(4), 685–691.

Kim, J., & Konstantinou, K. (2001). Digital predistortion of wideband signals based on power amplifier model with memory. *Electronics Letters*, *37*(23), 1417 – 1418.

Lacy, S. L., & Bernstein, D. S. (2003). Identification of FIR Wiener systems with unknown, non-invertible, polynomial non-linearities. *International Journal of Control*, *76*(15), 1500–1507.

Lataire, J., Piga, D., & Tóth, R. (2014). Frequency-domain Least-Squares Support Vector Machines to deal with correlated errors when identifying linear time-varying systems. In *19th world congress of the international federation of* *automatic control* (pp. 10024–10029). Cape Town, South Africa.

Lataire, J., Pintelon, R., Piga, D., & Tóth, R. (2017). Continuous-time linear time-varying system identification with a frequency-domain kernel-based estimator. *IET Control Theory & Applications*, *11*, 457–465.

Laurain, V., Tóth, R., Piga, D., & Zheng, W. (2015). An instrumental Least Squares Support Vector Machine for nonlinear system identification. *Automatica*, *54*, 340–347.

Laurain, V., Zheng, W., & Tóth, R. (2011). Introducing instrumental variables in the LS-SVM based identification framework. In *Proceedings of the 50th IEEE conference on decision and control and European Control Conference (CDC-ECC)* (pp. 3198–3203). Orlando, FL, USA.

Lee, Y. J., Sung, S. W., Park, S., & Park, S. (2004). Input test signal design and parameter estimation method for the Hammerstein-Wiener processes. *Industrial & engineering chemistry research*, *43*(23), 7521–7530.

Li, X., Xie, C., He, Y., Qiu, Z., & Zhang, Y. (2012). Characterizing the moisture content of tea with diffuse reflectance spectroscopy using wavelet transform and multivariate analysis. *Sensors*, *12*, 9847–9861.

Lindsten, F., Schön, T. B., & Jordan, M. I. (2013). Bayesian semiparametric Wiener system identification. *Automatica*, *49*(7), 2053–2063.

Ljung, L. (1999). *System identification : theory for the user*. Upper Saddle River (NJ): Prentice Hall PTR.

Ljung, L., Zhang, Q., Lindskog, P., Iouditski, A., & Singh, R. (2007). An integrated system identification toolbox for linear and nonlinear models. In *Proceedings of the 4th IFAC symposium on system identification, Newcastle, Australia.*

Lu, Z., Sun, J., & Butts, K. (2017). Multiscale support vector learning with projection operator wavelet kernel for nonlinear dynamical system identification. *IEEE Transactions on Neural Networkds and Learning Systems*, *28*, 231–243.

MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, *4*(3), 415–447.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics*, *15*(4), 661–675.

Marconato, A., Schoukens, M., Rolain, Y., & Schoukens, J. (2013). Study of the effective number of parameters in nonlinear identification benchmarks. In *Proceedings of the 52nd IEEE conference on decision and control (CDC 2013)* (pp. 4308–4313).

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 415–446.

Moody, J. E. (1991). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In *NIPS* (Vol. 4, pp. 847–854).

Münker, T., & Nelles, O. (2016). Nonlinear system identification with regularized local FIR model networks. *IFAC-PapersOnLine (4th IFAC Conference on Intelligent Control and Automation Sciences (ICONS))*, *49*(5), 61–66.

Mzyk, G. (2007). A censored sample mean approach to nonparametric identification of nonlinearities in Wiener systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *54*(10), 897–901.

Mzyk, G. (2014). *Combined parametric-nonparametric identification of block-oriented systems*. Springer.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The computer journal*, *7*(4), 308–313.

Novak, A., Simon, L., Kadlec, F., & Lotton, P. (2010). Nonlinear system identification using exponential swept-sine signal. *IEEE Transactions on Instrumentation and Measurement*, *59*(8), 2220–2229.

Nyström, E. J. (1930). Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Mathematica*, *54*(1), 185–204.

Palanthandalam-Madapusi, H. J., Ridley, A. J., & Bernstein, D. S. (2005). Identification and prediction of ionospheric dynamics using a Hammerstein-Wiener model with radial basis functions. In *American Control Conference, 2005. Proceedings of the 2005* (pp. 5052–5057).

Park, H. C., Sung, S. W., & Lee, J. (2006). Modeling of Hammerstein-Wiener processes with special input test signals. *Industrial & engineering chemistry research*, *45*(3), 1029–1038.

Pelckmans, K. (2011). Minlip for the identification of monotone Wiener systems. *Automatica*, *47*(10), 2298–2305.

Pintelon, R., & Schoukens, J. (2012). *System identification: a frequency domain approach*. John Wiley & Sons.

Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, *78*(9), 1481–1497.

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian processes for machine learning*. MIT Press.

Ravaud, R., Lemarquand, G., & Roussel, T. (2009). Time-varying non linear modeling of electrodynamic loudspeakers. *Applied Acoustics*, *70*(3), 450–458.

Risuleo, R. S., Bottegal, G., & Hjalmarsson, H. (2015). A new kernel-based approach to overparameterized Hammerstein system identification. In *Proceedings of the 54th IEEE conference on decision and control (CDC 2015)* (pp. 115–120).

Ron, K., & Foster, P. (1998). Special issue on applications of machine learning and the knowledge discovery process. *Journal of Machine Learning*, *30*, 271–274.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proceedings of the fifteenth international conference on machine learning (ICML 1998)*. Morgan Kaufmann Publishers Inc.

Schoukens, J., Dobrowiecki, T., & Pintelon, R. (1998). Parametric and nonparametric identification of linear systems in the presence of nonlinear distortions-a frequency domain approach. *IEEE Transactions on Automatic Control*, *43*(2), 176–190.

Schoukens, J., Nemeth, J. G., Crama, P., Rolain, Y., & Pintelon, R. (2003). Fast approximate identification of nonlinear systems. *Automatica*, *39*(7), 1267–1274.

Schoukens, J., Pintelon, R., & Rolain, Y. (2012). *Mastering system identification in 100 exercises*. John Wiley & Sons.

Schoukens, J., Suykens, J. A. K., & Ljung, L. (2009). Wiener-Hammerstein benchmark. In *Proceedings of the 15th IFAC Symposium on System Identification.*

Schoukens, J., Tiels, K., & Schoukens, M. (2014). Generating initial estimates for Wiener-Hammerstein systems using phase coupled multisines. In *Proceedings of 19th IFAC World Congress.*

Schoukens, J., Widanage, W. D., Godfrey, K. R., & Pintelon, R. (2007). Initial estimates for the dynamics of a Hammerstein system. *Automatica*, *43*(7), 1296–1301.

Schoukens, M. (2015). *Identification of parallel block-oriented models starting from the best linear approximation* (PhD Dissertation). Vrije Universiteit Brussel, Leegstraat 15, B-9060 Zelzate.

Schoukens, M., & Tiels, K. (2016). Identification of block-oriented nonlinear systems starting from linear approximations: A survey. *Automatica (Accepted for publication).*

Seeger, M. (2004). Gaussian processes for machine learning. *International journal of neural systems*, *14*(02), 69–106.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., ... Juditsky, A. (1995). Nonlinear black-box modmodel in system identification: a unified overview. *Automatica*, *31*, 1691–1724.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(4), 583–639.

Srinivasan, R., Rengaswamy, R., Narasimhan, S., & Miller, R. (2005). Control loop performance assessment. 2. Hammerstein model approach for stiction diagnosis. *Industrial & engineering chemistry research*, *44*(17), 6719–6728.

Stapleton, J., & Bass, S. (1985). Adaptive noise cancellation for a class of nonlinear, dynamic reference channels. *IEEE transactions on circuits and systems*, *32*(2), 143–150.

Sun, L., Liu, W., & Sano, A. (1999). Identification of a dynamical system with input nonlinearity. In *Control theory and applications, iee proceedings-* (Vol. 146, pp. 41–51).

Sung, S. W. (2002). System identification method for Hammerstein processes. *Industrial & engineering chemistry research*, *41*(17), 4295–4302.

Suykens, J. A. K. (2001). Nonlinear modelling and support vector machines. In *18th IEEE Instrumentation and Measurement Technology Conference (IMTC).* Budapest, Hungary.

Suykens, J. A. K., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation.

Neurocomputing, Special issue on fundamental and information processing aspects of neurocomputing, *48*(1-4), 85–105.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*, 293–300.

Suykens, J. A. K., & Vandewalle, J. (2000). Recurrent least squares support vector machines. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, *47*, 1109–1114.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least Squares Support Vector Machines*. World Scientific.

Szegő, G. (1921). Über orthogonale polynome, die zu einer gegebenen kurve der komplexen ebene gehören. *Mathematische Zeitschrift*, *9*(3), 218–270.

Tötterman, S., & Toivonen, H. (2009). Support vector method for identification of Wiener models. *Journal of Process Control*, *19*, 1174–1181.

Vanbeylen, L., Pintelon, R., & Schoukens, J. (2008). Blind maximum likelihood identification of Hammerstein systems. *Automatica*, *44*(12), 3139–3146.

Vanbeylen, L., Pintelon, R., & Schoukens, J. (2009). Blind maximum-likelihood identification of Wiener systems. *IEEE Transactions on Signal Processing*, *57*(8), 3017–3029.

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems: Theory—implementation—applications*. Springer Science & Business Media.

Vapnik, V. (1998). The support vector method of function estimation. In J. A. K. Suykens & J. Vandewalle (Eds.), *Nonlinear modeling: Advanced black-box techniques* (pp. 55–85). Boston: Kluwer Academic Publishers.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley New York.

Verhaegen, M., & Westwick, D. (1996). Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, *63*(2), 331–349.

Wachel, P., & Mzyk, G. (2016). Direct identification of the linear block in Wiener system. *International Journal of Adaptive Control and Signal Processing*, *30*(1), 93–105.

Wahba, G. (1990). *Spline models for observational data* (Vol. 59). SIAM.

Wahlberg, B., Welsh, J., & Ljung, L. (2014). Identification of Wiener systems with process noise is a nonlinear errors-in-variables problem. In *Proceedings of the 53rd IEEE conference on decision and control (CDC 2014)* (pp. 3328–3333).

Wahlberg, B., Welsh, J., & Ljung, L. (2015). Identification of stochastic Wiener systems using indirect inference. *IFAC-PapersOnLine*, *48*(28), 620–625.

Wang, J., Sano, A., Chen, T., & Huang, B. (2009). Identification of Hammerstein systems without explicit parameterisation of non-linearity. *International Journal of Control*, *82*(5), 937–952.

Wang, Z., Zhang, Z., Mao, J., & Zhou, K. (2012). A Hammerstein-based model for rate-dependent hysteresis in piezoelectric actuator. In *Control and Decision Conference (CCDC), 2012 24th Chinese* (pp. 1391–1396).

Weinberger, K. Q., & Tesauro, G. (2007). Metric learning for kernel regression. In *International conference on artificial intelligence and statistics* (pp. 612–619).

Westwick, D., & Verhaegen, M. (1996). Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing*, *52*(2), 235–258.

Westwick, D. T., & Kearney, R. E. (2003). *Identification of nonlinear physiological systems* (Vol. 7). John Wiley & Sons.

Wiener, N. (1958). *Nonlinear problems in random theory*. Wiley.

Wigren, T. (1993). Recursive prediction error identification using the nonlinear Wiener model. *Automatica*, *29*(4), 1011–1025.

Wigren, T. (1994). Convergence analysis of recursive identification algorithms based on the nonlinear Wiener model. *IEEE Transactions on Automatic Control*, *39*(11), 2191–2206.

Williams, C. K., & Seeger, M. (2000). Using the Nyström method to speed up kernel machines. In *Proceedings of the 13th international conference on neural information processing systems* (pp. 661–667).

Xavier-de Souza, S., Suykens, J. A. K., Vandewalle, J., & Bollé, D. (2009). Coupled simulated annealing. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *40*(2), 320–335.

Xie, C., Wang, Q., & He, Y. (2014). Identification of different varieties of sesame oil using near-infrared hyperspectral imaging and chemometrics algorithms. *PLoS ONE*, *9*(5). (art. no. e98522) doi: 10.1371/journal.pone.0098522

Young, P. (2000). Stochastic, dynamic modelling and signal processing: time variable and state dependent parameter estimation. *Nonlinear and nonstationary signal processing*, 74–114.

Young, P. C., McKenna, P., & Bruun, J. (2001). Identification of non-linear stochastic systems by state dependent parameter estimation. *International Journal of Control*, *74*(18), 1837–1857.

Zhu, Y. (1999). Distillation column identification for control using wiener model. In *Proceedings of the american control conference* (Vol. 5, pp. 3462–3466).

Curriculum Vitae

Ricardo Castro-Garcia was born in Medellín, Colombia. He received a B.Sc. in Electronic Engineering from Universidad Pontificia Bolivariana in 2004. In 2010, he obtained a post graduate degree in industrial maintenance at EAFIT University and another one in project management from Universidad Pontificia Bolivariana in 2012. From 2003 to 2009 he occupied various positions as engineer at Crystal Hosiery group, and since 2009 until 2012 he joined Ross International group as the maintenance director of Andes International Tooling and Moldes Medellin companies.

In 2012 Ricardo moved to Leuven, Belgium, where he obtained a M.Sc. in artificial intelligence, from KU Leuven. Since 2013 to 2017 he worked on his Ph.D. degree in the Department of Electrical Engineering (ESAT-STADIUS) of KU Leuven under the supervision of professors Johan A.K. Suykens and Johan Schoukens (VUB).

Ricardo's research focus has been on Machine Learning applied to nonlinear system identification. In particular, he has used kernel methods to develop new and more powerful methodologies for the identification of Block Oriented Nonlinear systems. These methodologies include different reformulations of existing black-box algorithms to incorporate additional available information into models that otherwise would be solely driven by the relationship of the input and output signals.

List of publications

Published and accepted for publication

Castro-Garcia, R., Tiels, K., Agudelo, O. M., Suykens, J. A. K. (2017). Hammerstein System Identification through Best Linear Approximation Inversion and Regularization. International Journal of Control. doi: 10.1080/00207179.2017.1329550. Available online at http://www.tandfonline.com/doi/abs/ 10.1080/00207179.2017.1329550.

Castro-Garcia, R., Agudelo, O. M., Suykens, J. A. K. (2017a). Impulse response constrained LS-SVM modeling for MIMO Hammerstein system identification. International Journal of Control. doi: 10.1080/00207179.2017.1373862. Available online at http://www.tandfonline.com/doi/abs/10.1080/ 00207179.2017.1373862

Castro-Garcia, R., Agudelo, O. M., Suykens, J. A. K. (2017b). *Impulse Response Constrained LS-SVM modeling for Hammerstein System Identification.* In proceedings of the 20th world congress of the International Federation of Automatic Control (IFAC 2017), Toulouse, France. (pp. 14611 – 14616).

Castro-Garcia, R., Agudelo, O. M., Suykens, J. A. K. (2017c). *MIMO Hammerstein System Identification using LS-SVM and Steady State Time Response.* Accepted for publication in the proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI 2017). (Internal Report 17-23, ESAT-SISTA, KU Leuven. Leuven, Belgium).

Bottegal, G.,**Castro-Garcia, R.**, Suykens, J. A. K. (2017a). *On the identification of Wiener systems with polynomial nonlinearity.* Accepted for publication in the proceedings of the 56th IEEE Conference on Decision and Control (CDC 2017). (Internal Report 17-55, ESAT-SISTA, KU Leuven. Leuven, Belgium).

Castro-Garcia, R., Agudelo, O. M., Tiels, K., Suykens, J. A. (2016). *Hammerstein system identification using LS-SVM and steady state time response.* In proceedings of the 15th European Control Conference (pp. 1063 – 1068).

Castro-Garcia, R., Suykens, J. A. (2016). *Wiener System Identification using Best Linear Approximation within the LS-SVM framework.* In proceedings of the 3rd Latin American Conference on Computational Intelligence. doi:10.1109/LA-CCI.2016.7885698.

Castro-Garcia, R., Tiels, K., Schoukens, J., Suykens, J. A. K. (2015). *Incorporating Best Linear Approximation within LS-SVM-Based Hammerstein System Identification.* In proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015), Osaka, Japan. (pp. 7392 - 7397).

Castro, R., Mehrkanoon, S., Marconato, A., Schoukens, J., Suykens, J. (2014). *SVD truncation schemes for fixed-size kernel models.* In proceedings of the International Joint Conference on Neural Networks. IJCNN 2014. Beijing, China, Jun. 2014 (pp. 3922-3929).

Under revision

Bottegal, G.,**Castro-Garcia, R.**, Suykens, J. A. K. (2017b). *A two-experiment approach to Wiener system identification.* In Internal report 17-38, ESAT-SISTA, KU Leuven (Leuven, Belgium).

Castro-Garcia, R., Tiels, K., Suykens, J. A. K. (2017). *Frequency Division LS-SVM for Nonlinear Modeling.* In Internal report 17-24, ESAT-SISTA, KU Leuven (Leuven, Belgium).

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
ESAT-STADIUS
Kasteelpark Arenberg 10 - box 2446
B-3001 Leuven
ricardo.castro@esat.kuleuven.be
https://www.esat.kuleuven.be/