Control Engineering Practice 20 (2012) 1165–1174

Contents lists available at SciVerse ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Least-Squares Support Vector Machines for the identification of Wiener–Hammerstein systems

Tillmann Falck [a,b,*], Philippe Dreesen [a,b], Kris De Brabanter [a,b], Kristiaan Pelckmans [c], Bart De Moor [a,b], Johan A.K. Suykens [a,b]

[a] KU Leuven, Department of Electrical Engineering ESAT/SCD, Kasteelpark Arenberg 10/2446, B-3001 Leuven, Belgium
[b] KU Leuven, IBBT Future Health Department, Leuven, Belgium
[c] Uppsala University, Division of Systems and Control, Department of Information Technology, SE-751 05 Uppsala, Sweden

ARTICLE INFO

Article history:
Received 7 March 2011
Accepted 9 May 2012
Available online 7 June 2012

Keywords:
Nonlinear system identification
LS-SVMs
Kernel-based models
Overparameterization
Large-scale data processing

ABSTRACT

This paper considers the identification of Wiener–Hammerstein systems using Least-Squares Support Vector Machines based models. The power of fully black-box NARX-type models is evaluated and compared with models incorporating information about the structure of the systems. For the NARX models it is shown how to extend the kernel-based estimator to large data sets. For the structured model the emphasis is on preserving the convexity of the estimation problem through a suitable relaxation of the original problem. To develop an empirical understanding of the implications of the different model design choices, all considered models are compared on an artificial system under a number of different experimental conditions. The obtained results are then validated on the Wiener–Hammerstein benchmark data set and the final models are presented. It is illustrated that black-box models are a suitable technique for the identification of Wiener–Hammerstein systems. The incorporation of structural information results in significant improvements in modeling performance.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Wiener–Hammerstein systems (Billings & Fakhouri, 1978) are a particular class of structured nonlinear systems. They consist of a static nonlinear function that is sandwiched between a linear system at the input and another linear system at the output as shown in Fig. 1. There is extensive literature on the identification of block structured nonlinear systems (see e.g., Giri & Bai, 2010). The identification of Wiener–Hammerstein systems received attention in Bershad, Celka, and Mclaughlin (2001), Boutayeb and Darouach (1995), Enqvist and Ljung (2005), Goodman, Herman, Bond, and Miller (2009), Greblicki and Pawlak (2008), Kibangou and Favier (2006), and Tan and Godfrey (2002), and saw an increase of activity since a dedicated special session at SYSID2009 (Schoukens, Suykens, & Ljung, 2009).

The scope of this paper is to study how well black-box modeling approaches (Sjöberg et al., 1995) are suited to obtain good models for systems of Wiener–Hammerstein structure. This is done for a complete black-box NARX model structure (Bai & Liu, 2007; Leontaritis & Billings, 1985; Lin, Horne, Tino, & Giles, 1996; Lind & Ljung, 2008; Pillonetto, Quang, & Chiuso, 2011; Piroddi & Spinelli, 2003; Roll, Nazin, & Ljung, 2005; Spinelli, Piroddi, & Lovera, 2005) as well as using a model structure tailored to Wiener–Hammerstein systems. Two sets of data are considered. The first is created artificially and represents different experimental conditions such as the number of measurements and signal to noise ratio. The second data set is the Wiener–Hammerstein benchmark data set (Schoukens et al., 2009) which contains the measurements of a nonlinear electrical circuit.

Kernel-based methods, including smoothing splines (Wahba, 1990), Support Vector Machines (SVMs) (Schölkopf & Smola, 2002; Vapnik, 1998) and Gaussian Processes (Rasmussen & Williams, 2006), have shown to be powerful techniques in nonlinear regression. One of the reasons for many recent identification methods to adopt a kernel framework is the simplicity of selecting basis functions. In all these approaches only a single kernel function has to be chosen, for which the popular Gaussian RBF kernel performs very well on most problems. Alternatives like the polynomial kernel and special kernels for dynamical systems (Pillonetto et al., 2011) are less common in practice, but can yield better results in some cases. The polynomial kernel relates to polynomial expansions, which are known to suffer from instabilities in many cases. By adopting the kernel framework, the estimation problem can be formulated in the dual where the explicit exponential growth of monomial terms is circumvented, and instead the number of parameters is determined by the size of the

* Corresponding author at: KU Leuven, Department of Electrical Engineering ESAT/SCD, Kasteelpark Arenberg 10/2446, B-3001 Leuven, Belgium.
E-mail addresses: tillmann.falck@esat.kuleuven.be (T. Falck), philippe.dreesen@esat.kuleuven.be (P. Dreesen), kris.debrabanter@esat.kuleuven.be (K. De Brabanter), kristiaan.pelckmans@it.uu.se (K. Pelckmans), bart.demoor@esat.kuleuven.be (B. De Moor), johan.suykens@esat.kuleuven.be (J.A.K. Suykens).

0967-0661/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.conengprac.2012.05.006

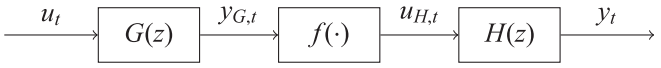


Fig. 1. General structure of a Wiener–Hammerstein system with measured input signal u_t and output signal y_t. The intermediate signals $y_{G,t}$ and $u_{H,t}$ are assumed to be unobserved.

data set. Another main advantage of many kernel-based estimation techniques is the presence of inherent regularization providing effective complexity control even when operating in high dimensional input spaces.

For this paper Least-Squares Support Vector Machines (LS-SVMs) (Suykens, Van Gestel, De Brabanter, De Moor, & Vandewalle, 2002, 2010) are used. It has been shown that LS-SVMs work well in practice for system identification problems (see e.g., Espinoza, Suykens, Belmans, & De Moor (2007)). Least-Squares Support Vector techniques perform nonlinear regression by projecting the input variables into a high (possibly infinite) dimensional feature space and then solve a regularized linear regression problem in this so-called feature space. Adopting LS-SVMs as base model for the modeling of nonlinear systems has several advantages. As for many other kernel-based estimation techniques, inherent regularization provides effective complexity control. Both the optimal model representation and the estimation problem follow from the conditions for optimality. The estimation of LS-SVMs is numerically inexpensive as it only requires the solution of linear systems. Except for the selection of a few tuning parameters the estimation problem for support vector approaches is convex and can be solved exactly. For the selection of the tuning parameters (e.g., kernel parameters and regularization constant) many different approaches have been proposed in the context of SVMs, but in most cases good performance can be achieved using straightforward strategies like cross-validation.

One of the key advantages of LS-SVMs utilized here is its problem statement as an optimization problem with primal and dual formulations. Initially LS-SVMs are formulated as a parametric and finite dimensional optimization problem subject to linear equality constraints. This form is called primal formulation and not only easy to comprehend but furthermore offers straightforward and efficient ways to modify the base model. The actual kernel-based representation is then achieved by exploiting Lagrangian duality and deriving the dual problem in terms of the kernel function. This paper considers two modifications explicitly which make use of both, the primal and the dual formulation, of LS-SVMs. Both extensions of the base LS-SVM model are briefly outlined in the following paragraphs.

One challenge in kernel-based techniques is handling large data sets as they are commonly found in system identification tasks. The kernel matrix scales quadratically with the number of data and therefore limits the amount of data that can be processed directly. However, in addition to the dual formulation in terms of the kernel matrix, LS-SVMs possess the primal description expressed in terms of the feature map. A finite dimensional approximation of this feature map can be obtained from a subsample of the complete kernel matrix. This then allows solving the primal problem for very large data sets. This approach is known as fixed-size LS-SVM (Suykens et al., 2002) and has been successfully applied for system identification (Espinoza et al., 2007) of Wiener–Hammerstein systems (De Brabanter et al., 2009).

For improved prediction performance, models that exploit knowledge about the true system, like its structure, are helpful. Therefore in addition to fully black-box NARX models, models incorporating structural information are considered. Goethals, Pelckmans, Suykens, and De Moor (2005, 2010, chap. 15) describe a convex approximation of Hammerstein systems based on the LS-SVM formulation. The idea is based on overparameterization (Bai, 1998; Chang & Luus, 1971) where nonconvex terms are replaced by new variables and relies on the possibility to make modifications to the primal problem formulation, especially the set of constraints. The newly found primal formulation is then straightforwardly transferred to a kernel-based formulation in the dual relying on Lagrangian duality. The procedure is extended to Wiener–Hammerstein systems by jointly modeling the input linear system $G(z)$ and the nonlinearity $f(\cdot)$ (Falck, Pelckmans, Suykens, & De Moor, 2009).

The paper starts with a brief explanation of LS-SVMs in Section 2. Section 3 discusses how large data sets can be handled by approximating the feature map. Section 4 illustrates how structural information can be integrated into an LS-SVM formulation. Numerical results for different experimental conditions on an artificial data set are reported in Section 5. Finally Section 6 evaluates the presented techniques on the Wiener–Hammerstein benchmark data set.

2. LS-SVMs for system identification

2.1. The NARX model structure

The most common model structure used for nonlinear SISO system identification is a NARX model (Sjöberg et al., 1995) of the form

$$y_t = f(y_{t-1},\ldots,y_{t-p},u_t,u_{t-1},\ldots,u_{t-q}) + e_t, \tag{1}$$

where $y_t \in \mathbb{R}$ and $u_t \in \mathbb{R}$ denote the measured output and the measured input at time instance t, respectively, and $f: \mathbb{R}^{p+q+1} \to \mathbb{R}$ is an unknown function. The observation noise e_t is assumed to be i.i.d. with zero mean and finite variance. The order of the NARX model is determined by the input and output orders p and q, representing the number of lags. In this way, the NARX model is defined as a nonlinear regressor by estimating in each step the next output value based on the past input and output measurements, i.e., $y_t = f(\boldsymbol{x}_t) + e_t$, where $\boldsymbol{x}_t$ is the vector of regressors containing the past input and output data.

2.2. Least-Squares Support Vector Machines

In this paper the function f in (1) is modeled as a Least-Squares Support Vector Machine (Suykens et al., 2002, 2010). Support vector machines formulations are set within the context of convex optimization theory and statistical learning. The general methodology is to phrase the modeling problem in the so-called *primal weight space* as a constrained optimization problem. In the next step the Lagrangian is formulated, and the conditions for optimality are obtained to state the problem in the *dual space* in terms of Lagrange multipliers.

Given a data set $\{(\boldsymbol{x}_t,y_t)\}_{t=D}^{N}$ where $D = \max(p,q)+1$, the goal is to estimate a model of the form

$$y_t = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_t) + c + e_t, \tag{2}$$

where $\boldsymbol{x}_t \in \mathbb{R}^{p+q+1}$ represents the $p+q+1$-dimensional input data, $y_t \in \mathbb{R}$ represents the scalar valued outputs, and the function $\boldsymbol{\varphi}: \mathbb{R}^{p+q+1} \to \mathbb{R}^{n_h}$ denotes the feature map from the $p+q+1$-dimensional input space to a high-dimensional (and possibly infinite dimensional) feature space (Vapnik, 1998). The error terms e_t are assumed to be i.i.d. with zero mean and finite variance. The unknowns in the model are $\boldsymbol{w} \in \mathbb{R}^{n_h}$ and $c \in \mathbb{R}$.

The unknowns $\boldsymbol{w}$ and c are determined from a convex optimization problem which is formulated as

$$\begin{aligned} &\min_{\boldsymbol{w},c,e_t} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{2}\gamma\sum_{t=D}^{N} e_t^2, \\ &\text{subject to} \quad y_t = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_t) + c + e_t, \quad t = D,\ldots,N, \end{aligned} \tag{3}$$

where γ is a regularization constant trading fitting accuracy against model complexity.

Lemma 1 (*Suykens et al., 2002*). *After the elimination of* $\boldsymbol{w}$ *and* $\boldsymbol{e}$, *the solution to* (3) *is given by the dual problem*

$$\begin{bmatrix} \boldsymbol{\Omega}+\gamma^{-1}\mathbf{I}_{\tilde{N}} & \mathbf{1}_{\tilde{N}} \\ \mathbf{1}_{\tilde{N}}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ c \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}, \tag{4}$$

with $\boldsymbol{y}=[y_D,\ldots,y_N]^T$, $\mathbf{1}_{\tilde{N}}=[1,\ldots,1]^T\in\mathbb{R}^{\tilde{N}}$, $\tilde{N}=N-D+1$, *and* $\boldsymbol{\alpha}$ *is the vector containing the Lagrange multipliers,* $\boldsymbol{\alpha}=[\alpha_1,\ldots,\alpha_{\tilde{N}}]^T$. *The kernel matrix is given by* $\Omega_{kl}=\boldsymbol{\varphi}(\boldsymbol{x}_k)^T\boldsymbol{\varphi}(\boldsymbol{x}_l)=K(\boldsymbol{x}_k,\boldsymbol{x}_l)$, *for* $k,l=1,\ldots,\tilde{N}$.

The LS-SVM model is expressed in dual form as

$$\hat{y}(\boldsymbol{z})=\sum_{t=1}^{\tilde{N}}\alpha_t K(\boldsymbol{z},\boldsymbol{x}_t)+c. \tag{5}$$

In general, the feature space can be high-dimensional (and possibly infinite-dimensional) and is only implicitly defined by the kernel. However, note that in the dual formulation, it is not required to evaluate $\boldsymbol{w}$ and $\boldsymbol{\varphi}(\cdot)$ explicitly, as the required operations are achieved implicitly through the use of positive definite kernel functions K. For the kernel function K there are typically a number of choices: (1) linear kernel: $K(\boldsymbol{x}_k,\boldsymbol{x}_l)=\boldsymbol{x}_k^T\boldsymbol{x}_l$; (2) polynomial kernel with degree s, and a tuning parameter $r\geq 0$: $K(\boldsymbol{x}_k,\boldsymbol{x}_l)=(\boldsymbol{x}_k^T\boldsymbol{x}_l+r)^s$; (3) Gaussian radial basis function (RBF) kernel with bandwidth σ: $K(\boldsymbol{x}_k,\boldsymbol{x}_l)=\exp(-\|\boldsymbol{x}_k-\boldsymbol{x}_l\|^2/\sigma^2)$. For all experiments described in this paper, the Gaussian RBF kernel will be used. Another possible choice for the kernel function is e.g., a polynomial kernel, but the Gaussian RBF kernel usually performs best. A comparison of these two kernels for the benchmark data set considered later on and employing the NARX structure is given in De Brabanter et al. (2009).

In practice, most effort in training the LS-SVM model is spent on the optimal selection of the tuning parameters, e.g., σ and γ in case a Gaussian RBF kernel is used, which is usually obtained by validation on an independent segment of the data set, cross-validation, Bayesian inference, bootstrap methods, and others. The tuning parameters are in this case determined by means of a grid search: a grid is fixed over a large range of σ and γ values. Next, the optimal parameters minimizing the performance on an independent validation set are chosen. Other approaches to obtain suitable σ and γ values are possible as well, such as evolutionary algorithms, (coupled) simulated annealing, or the simplex method (De Brabanter et al., 2010). After the tuning parameters are chosen, c and $\boldsymbol{\alpha}$ are found from solving a linear system. The necessary computations therefore are summarized in Algorithm 1. The complete process of model selection is outlined in Algorithm 2 and applies not only to the most commonly used grid search procedure but also the more sophisticated approaches mentioned above.

Algorithm 1. Estimation of basic LS-SVM model.

1. Select models orders p and q.
2. Select a regularization parameter γ and a kernel function K (and its parameters).
3. Compute the kernel matrix $\boldsymbol{\Omega}$.
4. Solve the dual linear system (4).

Algorithm 2. Model selection for kernel-based model.

1. Pick nominal values for the model orders p,q, the regularization parameter γ, the kernel function K (and its parameters).
2. Estimate a single model with Algorithm 1 on a subset of the measured data to obtain model parameter $\boldsymbol{\alpha}$, c.
3. Predict values using this model by evaluating (5) on a different subset of the measured data.
4. Compute a cost function, e.g., the root mean squared error (RMSE) comparing the predicted with the measured values.
5. If cost is small enough, stop; otherwise change one of the parameters and repeat from 2.

Remark 1. The computational cost required for the methods presented in this work is determined by the solution of (large) systems of linear equations, as in (4) and (21) introduced later in this paper. The worst-case complexity is $O(N^3)$ where N is the number of training samples. Note that this is independent of the dimension of the input space. In comparison with polynomial expansions in parametric models this is a considerable advantage. In case the number of samples is very large, as it is the case for the benchmark data set considered towards the end of the paper, the complexity can be reduced to $O(M^3)$ where M is usually much smaller than the number of samples N and denotes the approximation dimension of the feature map $\boldsymbol{\varphi}$. This will be explained in more detail in the following section with the corresponding linear systems being defined by (10) and (24). Therefore the individual cost of estimating a single model can be kept very moderate, however the total cost can grow depending on how many models are evaluated for model selection.

3. Handling large data sets

Solving the LS-SVM modeling problem in the dual space circumvents the explicit formulation in the high-dimensional feature space (De Brabanter, De Brabanter, Suykens, & De Moor, 2010; Espinoza et al., 2007; Suykens et al., 2002). Instead it involves solving the square linear system (4) of size $(\tilde{N}+1)\times(\tilde{N}+1)$. If the number of available data points $\tilde{N}$ is very large, this can be computationally infeasible. In such cases, it is more interesting to work directly in the primal space, however then an explicit expression of the feature map is needed. For these cases techniques such as ridge regression for parametric estimation in the primal space can be used. In this section, the estimation problem in primal space is described in terms of an explicit approximation of the nonlinear mapping $\boldsymbol{\varphi}(\cdot)$, and the further implementation for a large-scale problem.

3.1. Nyström approximation of the feature map

Given a data set $\{(\boldsymbol{x}_t,y_t)\}_{t=D}^N$ where $D=\max(p,q)+1$, a finite number of eigenfunctions $\varphi_i(\boldsymbol{x})$ of the kernel can be obtained as follows (Williams & Seeger, 2001). Consider the Fredholm integral equation of the first kind function, i.e.,

$$\int K(\boldsymbol{x},\boldsymbol{z})\varphi_i(\boldsymbol{x})p(\boldsymbol{x})\,d\boldsymbol{x}=\lambda_i\varphi_i(\boldsymbol{z}), \tag{6}$$

with solutions λ_i and $\varphi_i(\boldsymbol{x})$, $i=1,\ldots,n_h$, for a variable $\boldsymbol{x}$ with probability density $p(\boldsymbol{x})$. Therefore one has $\boldsymbol{\varphi}(\boldsymbol{x})=[\sqrt{\lambda_1}\varphi_1(\boldsymbol{x}),\ \sqrt{\lambda_2}\varphi_2(\boldsymbol{x}),\ldots,\sqrt{\lambda_{n_h}}\varphi_{n_h}(\boldsymbol{x})]^T$. The integral (6) can be approximated by means of the sample average. The eigenvalue decomposition of the kernel matrix $\boldsymbol{\Omega}=\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T$, with $\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_{\tilde{N}}$ and $\boldsymbol{\Sigma}$ diagonal, can then be used in order to compute the $\tilde{N}$-dimensional approximation of the feature map by means of

$$\hat{\boldsymbol{\varphi}}(\boldsymbol{z})=\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{k}(\boldsymbol{z}), \tag{7}$$

where $\boldsymbol{k}(\boldsymbol{z})=[K(\boldsymbol{x}_D,\boldsymbol{z}),\ldots,K(\boldsymbol{x}_N,\boldsymbol{z})]^T$. Suykens et al. (2002) then use the explicit finite-dimensional representation of the feature map $\hat{\boldsymbol{\varphi}}$ in the primal model formulation (3) to estimate $\boldsymbol{w}$ and c directly.

3.2. Sparse approximations and large-scale problems

The use of the entire training sample of size $\tilde{N}$ to compute the approximation of $\boldsymbol{\varphi}$ will produce a vector $\hat{\boldsymbol{\varphi}}(\boldsymbol{x})$ having $\tilde{N}$ components, each one of which can be computed by (7). For a large-scale problem, it has been motivated in Williams and Seeger (2001) to use a subsample of $M \ll \tilde{N}$ data points to compute $\hat{\boldsymbol{\varphi}}$. In this case, up to M components will have to be computed. In contrast to a random selection of the subsample, a procedure for active selection of the initial set of support vectors is proposed in Suykens et al. (2002). External criteria, such as entropy maximization, can be applied for an optimal selection of the subsample. In this case, the M support vectors are selected such that the quadratic Rényi entropy H_R is maximized, given as

$$H_R = -\log \int p^2(\boldsymbol{x})\, d\boldsymbol{x}, \tag{8}$$

with p the density of the selected support vectors. The quadratic entropy can be approximated on a finite subsample of size M (Girolami, 2002) using

$$\int \hat{p}^2(\boldsymbol{x})\, d\boldsymbol{x} \simeq \frac{1}{M^2} \mathbf{1}_M^T \boldsymbol{\Omega}_M \mathbf{1}_M, \tag{9}$$

where $\mathbf{1}_M = [1,\ldots,1]^T \in \mathbb{R}^M$ and $\boldsymbol{\Omega}_M = K(\boldsymbol{x}_k,\boldsymbol{x}_l)$, for $\boldsymbol{x}_k,\boldsymbol{x}_l \in S_M$, with $S_M \subset \{\boldsymbol{x}_t\}_{t=D}^N$ and $|S_M| = M$. The initial choices for the kernel function and its corresponding bandwidth used in this procedure are addressed in detail in De Brabanter et al. (2010). The use of this active selection procedure can be quite important for large-scale problems. The optimality of the selection is related to the final accuracy that can be obtained in the modeling problem. The use of an entropy maximization criterion ensures that the selected subsample is well spread over the entire data range and it will not be concentrated in a certain area of the data set.

3.3. Fixed-size LS-SVM

Based on the explicit approximation $\hat{\boldsymbol{\varphi}}$ that can be computed from an initial sample of M data points from the given data set $\{(\boldsymbol{x}_t,y_t)\}_{t=D}^N$, the fixed-size LS-SVM (FS-LSSVM) nonlinear regression can be formulated as the optimization problem (3) where the feature map $\boldsymbol{\varphi}$ in the constraints is replaced by its approximation $\hat{\boldsymbol{\varphi}}$. It is important to observe that this model is nonlinear in the inputs, but linear in the parameters $\tilde{\boldsymbol{w}}$ and $\check{c}$. Therefore its solution is determined by the linear system

$$\begin{bmatrix} \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \gamma^{-1}\mathbf{I}_M & \boldsymbol{\Phi}\mathbf{1}_{\tilde{N}} \\ \mathbf{1}_{\tilde{N}}^T\boldsymbol{\Phi}^T & \tilde{N} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{w}} \\ \check{c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^T\boldsymbol{y} \\ \mathbf{1}_{\tilde{N}}\boldsymbol{y} \end{bmatrix}, \tag{10}$$

where $\boldsymbol{\Phi} = [\hat{\boldsymbol{\varphi}}(\boldsymbol{x}_D),\ldots,\hat{\boldsymbol{\varphi}}(\boldsymbol{x}_N)]$. Note that this system is $(M+1)\times(M+1)$ rather than $(\tilde{N}+1)\times(\tilde{N}+1)$ as in (4). The model can be evaluated with

$$\hat{y}(\boldsymbol{z}) = \tilde{\boldsymbol{w}}^T\hat{\boldsymbol{\varphi}}(\boldsymbol{z}) + \check{c}. \tag{11}$$

A complete algorithm is given to estimate a FS-LSSVM model.

Algorithm 3. Estimation of FS-LSSVM model.

1. Select a subset $S_M \subset \{(\boldsymbol{x}_t,y_t)\}_{t=D}^N$ of $M \ll \tilde{N}$ data points from the data set.
2. Select a regularization parameter γ and a kernel function K (and its parameters).
3. Build the kernel matrix $\boldsymbol{\Omega}_M$ evaluated on S_M and compute its eigendecomposition $\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T$.
4. Use (7) to form $\boldsymbol{\Phi}$ by evaluating $\hat{\boldsymbol{\varphi}}$ for $\boldsymbol{x}_D,\ldots,\boldsymbol{x}_N$.
5. Solve the primal problem (3) via (10).

4. Exploiting information on the model structure

So far the Wiener–Hammerstein system shown in Fig. 1 is modeled with a NARX structure. This structure applies to a wide range of nonlinear systems and does not incorporate any prior knowledge on the particular structure of a Wiener–Hammerstein system. The current section is based on Falck et al. (2009) and demonstrates how to integrate structural information in an LS-SVM model. It is an extension of the overparameterization approach for Hammerstein systems (Bai, 1998; Chang & Luus, 1971) applied to LS-SVMs (Goethals et al., 2005, 2010, chap. 15).

4.1. Model parameterization and nonlinear estimation problem

In order to include structural information let the first linear block $G(z)$ have p_G poles and q_G zeros and the second linear block $H(z)$, p_H poles and q_H zeros, respectively. Let $y_{G,t}$ be the unobserved output of the linear system $G(z)$ and $u_{H,t}$ the unobserved input of the linear system $H(z)$ as seen in Fig. 1. Assuming linear ARX structures for both systems their output signals can be written as

$$y_{G,t} = \sum_{k=0}^{q_G} b_{G,k}u_{t-k} - \sum_{k=1}^{p_G} a_{G,k}y_{G,t-k}, \tag{12}$$

$$\hat{y}_t = \sum_{k=0}^{q_H} b_{H,k}u_{H,t-k} - \sum_{k=1}^{p_H} a_{H,k}y_{t-k}, \tag{13}$$

with parameters $\{a_{G,k}\}_{k=1}^{p_G}$ and $\{b_{G,k}\}_{k=0}^{q_G}$ for $G(z)$ and $\{a_{H,k}\}_{k=1}^{p_H}$ and $\{b_{H,k}\}_{k=0}^{q_H}$ for $H(z)$, respectively. Finally one can relate the output $y_{G,t}$ of $G(z)$ and the input $u_{H,t}$ of $H(z)$ through the static nonlinearity, such that

$$u_{H,t} = f(y_{G,t}). \tag{14}$$

As parameterization for $f(\cdot)$ an LS-SVM structure is chosen as explained in Section 2. Therefore one can express the nonlinearity as

$$f(x) = \boldsymbol{w}^T\boldsymbol{\varphi}(x) + c. \tag{15}$$

Note that in contrast with Section 2 the regression variable x is scalar and accordingly $\boldsymbol{\varphi}: \mathbb{R} \to \mathbb{R}^{n_h}$. To enable a compact notation let $\boldsymbol{a}_G = [a_{G,1},\ldots,a_{G,p_G}]^T \in \mathbb{R}^{p_G}$, $\boldsymbol{b}_G = [b_{G,0},\ldots,b_{G,q_G}]^T \in \mathbb{R}^{q_G+1}$, $\boldsymbol{a}_H = [a_{H,1},\ldots,a_{H,p_H}]^T \in \mathbb{R}^{p_H}$ and $\boldsymbol{b}_H = [b_{H,0},\ldots,b_{H,q_H}]^T \in \mathbb{R}^{q_H+1}$. Also define vectors of lagged inputs and outputs as $\boldsymbol{u}_t = [u_t,\ldots,u_{t-q_G}]^T \in \mathbb{R}^{q_G+1}$, $\boldsymbol{y}_{t-1} = [y_{t-1},\ldots,y_{t-p_H}] \in \mathbb{R}^{p_H}$ and similarly for the unobserved intermediate signals $\boldsymbol{y}_{G,t-1} = [y_{G,t-1},\ldots,y_{G,t-p_G}]^T \in \mathbb{R}^{p_G}$ and $\boldsymbol{u}_{H,t} = [u_{H,t},\ldots,u_{H,t-q_H}]^T \in \mathbb{R}^{q_H+1}$. Then all parameters can be estimated from a nonlinear least squares problem

$$\begin{aligned} \min_{\boldsymbol{a}_G,\boldsymbol{b}_G,\boldsymbol{a}_H,\boldsymbol{b}_H,\boldsymbol{w},c,y_{G,t},u_{H,t},e_t} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{2}\gamma\sum_{t=D_H}^{N} e_t^2, \\ \text{subject to} \quad & y_t = \boldsymbol{b}_H^T\boldsymbol{u}_{H,t} - \boldsymbol{a}_H^T\boldsymbol{y}_{t-1} + e_t,\ t = D_H,\ldots,N, \\ & u_{H,t} = \boldsymbol{w}^T\boldsymbol{\varphi}(y_{G,t}) + c, \quad t = D_G,\ldots,N, \\ & y_{G,t} = \boldsymbol{b}_G^T\boldsymbol{u}_t - \boldsymbol{a}_G^T\boldsymbol{y}_{G,t-1}, \quad t = D_G,\ldots,N, \\ & \|\boldsymbol{b}_G\|_2 = 1, \quad \|\boldsymbol{b}_H\|_2 = 1, \end{aligned} \tag{16}$$

with $D_G = \max(p_G,q_G)+1$ and $D_H = \max(p_H+1,q_H+D_G)$. The last two scaling constraints remove an ambiguity, as a constant gain can be shuffled between the linear blocks $G(z)$ and $H(z)$ and the nonlinearity $f(\cdot)$ (Boyd & Chua, 1983, 1985).

4.2. Overparameterization of a simplified model

As problem (16) is nonconvex, the model is simplified and rewritten in this subsection. The modified model then enables a convex relaxation. The procedure first eliminates references to

the unobserved signals $y_{G,t}$ and $u_{H,t}$ from (16). In a second step all remaining nonconvexity is captured in a rank constraint.

In order to eliminate the dependency on $y_{G,t}$ one can jointly model the first linear block $G(z)$ and the static nonlinearity $f(\cdot)$ with a NFIR structure. Therefore the static variable x in (15) is replaced by a vector of lagged inputs $\boldsymbol{u}_{f,t}=[u_t,\ldots,u_{t-q_f}]^T\in\mathbb{R}^{q_f+1}$. Here q_f has to be chosen large enough to approximate the impulse response of $G(z)$. Then the joint model can be written as

$$u_{H,t}=f(\boldsymbol{u}_{f,t})=\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{u}_{f,t})+c, \tag{17}$$

where now $\boldsymbol{\varphi}:\mathbb{R}^{q_f}\to\mathbb{R}^{n_h}$. Note that the modified model still depends on $u_{H,t}$. To also eliminate this signal from the modified model, one can substitute (17) into the model (13) for $H(z)$ to obtain

$$y_t=\sum_{k=0}^{q_H} b_{H,k}(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{u}_{f,t-k})+c)-\boldsymbol{a}_H^T\boldsymbol{y}_{t-1}+e_t. \tag{18}$$

Note that now the only remaining nonconvexity of the problem is due to the bilinear products of $b_{H,k}$ with $\boldsymbol{w}$ and c. The latter one is straightforwardly removed by introducing a new variable d with $d=c\sum_{k=0}^{q_H}b_{H,k}$. For the former let $\boldsymbol{\Phi}_t=[\boldsymbol{\varphi}(\boldsymbol{u}_{f,t}),\ldots,\boldsymbol{\varphi}(\boldsymbol{u}_{f,t-q_H})]$, then the sum can be rewritten as $\mathrm{tr}(\boldsymbol{b}_H^T\boldsymbol{\Phi}_t^T\boldsymbol{w})=\mathrm{tr}(\boldsymbol{\Phi}^T\boldsymbol{w}\boldsymbol{b}_H^T)$. Now the rank-1 product $\boldsymbol{w}\boldsymbol{b}_H^T$ can be replaced by $\boldsymbol{W}$ and the additional constraint $\mathrm{rank}(\boldsymbol{W})=1$. This leads to a new optimization problem

$$\begin{aligned}\min_{\boldsymbol{W},d,\boldsymbol{a}_H,e_t}\quad & \frac{1}{2}\|\boldsymbol{W}\|_F^2+\frac{1}{2}\gamma\sum_{t=D}^{N}e_t^2,\\ \text{subject to}\quad & y_t=\mathrm{tr}(\boldsymbol{\Phi}_t^T\boldsymbol{W})+d-\boldsymbol{a}_H^T\boldsymbol{y}_{t-1}+e_t,\ t=D,\ldots,N,\\ & \mathrm{rank}(\boldsymbol{W})=1,\end{aligned} \tag{19}$$

with $D=\max(p_H,q_f+q_H)+1$. The constraints are a reformulation of (18). This is also the case for the regularization term $\boldsymbol{w}^T\boldsymbol{w}$ of (16). Therefore it can be rewritten as $\mathrm{tr}(\boldsymbol{w}\boldsymbol{w}^T)$. Due to the scaling constraints $\boldsymbol{b}_H^T\boldsymbol{b}_H=1$. Hence, $\boldsymbol{w}^T\boldsymbol{w}=\mathrm{tr}(\boldsymbol{w}(\boldsymbol{b}_H^T\boldsymbol{b}_H)\boldsymbol{w}^T)=\mathrm{tr}(\boldsymbol{W}\boldsymbol{W}^T)=\|\boldsymbol{W}\|_F^2$.

4.3. Convex relaxation and dual model representation

The reformulated problem (19) is convex except for the rank-1 constraint. Therefore a convex approximation can be straightforwardly obtained by dropping the rank constraint. Goethals et al. (2005, 2010, chap. 15) note that model performance can be improved by centering the nonlinear contributions $\boldsymbol{w}_k^T\boldsymbol{\Phi}_{(k)}$ for $k=0,\ldots,q_H$, where $\boldsymbol{W}=[\boldsymbol{w}_0,\ldots,\boldsymbol{w}_{q_H}]$, $\boldsymbol{\Phi}_{(k)}=[\boldsymbol{\varphi}(\boldsymbol{u}_{f,D-q_H+k}),\ldots,\boldsymbol{\varphi}(\boldsymbol{u}_{f,N-q_H+k})]$ and $\tilde{N}=N-D+1$. This can be ensured by the introduction of new constraints $\boldsymbol{w}_k^T\boldsymbol{\Phi}_{(k)}\boldsymbol{1}_{\tilde{N}}=0$ for $k=0,\ldots,q_H$. For a more elaborate interpretation of this constraint the reader is referred to Falck, Signoretto, Suykens, and De Moor (2011). Note that $\|\boldsymbol{W}\|_F^2=\sum_{p=0}^{q_H}\boldsymbol{w}_k^T\boldsymbol{w}_k$, then the modified convex relaxation of (19) is given by

$$\begin{aligned}\min_{\boldsymbol{W},d,\boldsymbol{a}_H,e_t}\quad & \frac{1}{2}\sum_{k=0}^{q_h}\boldsymbol{w}_k^T\boldsymbol{w}_k+\frac{1}{2}\gamma\sum_{t=D}^{N}e_t^2,\\ \text{subject to}\quad & \boldsymbol{y}=\sum_{k=0}^{q_h}\boldsymbol{\Phi}_{(k)}^T\boldsymbol{w}_k+\boldsymbol{1}_{\tilde{N}}d-\boldsymbol{Y}^T\boldsymbol{a}_H+\boldsymbol{e},\\ & \boldsymbol{w}_k^T\boldsymbol{\Phi}_{(k)}\boldsymbol{1}_{\tilde{N}}=0,\quad k=0,\ldots,q_H,\end{aligned} \tag{20}$$

where $\boldsymbol{y}=[y_D,\ldots,y_N]^T$, $\boldsymbol{e}=[e_D,\ldots,e_N]^T$ and $\boldsymbol{Y}=[\boldsymbol{y}_{D-1},\ldots,\boldsymbol{y}_{N-1}]$. As outlined in Section 2 problem (20) is often not directly solvable because the feature map might not be known explicitly and can be infinite dimensional. Therefore as in Section 2 a dual problem in terms of the kernel function and Lagrange multipliers is derived.

Lemma 2. *Let $\boldsymbol{\alpha}\in\mathbb{R}^{\tilde{N}}$ denote the Lagrange multipliers for the first equality constraint of* (20) *and $\boldsymbol{\beta}=[\beta_0,\ldots,\beta_{q_H}]$ denote the Lagrange multipliers for the centering constraints. Then the solution of* (20) *can be obtained in the dual from the linear system*

$$\begin{bmatrix}\boldsymbol{\Omega}_A+\frac{1}{\gamma}\boldsymbol{I}_{\tilde{N}} & \boldsymbol{\Omega}_C & \boldsymbol{1}_{\tilde{N}} & -\boldsymbol{Y}^T\\ \boldsymbol{\Omega}_C^T & \boldsymbol{\Omega}_D & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{1}_{\tilde{N}}^T & \boldsymbol{0} & 0 & \boldsymbol{0}\\ -\boldsymbol{Y} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\\ d\\ \boldsymbol{a}_H\end{bmatrix}=\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{0}\\ 0\\ \boldsymbol{0}\end{bmatrix}, \tag{21}$$

where $\boldsymbol{\Omega}_A=\sum_{k=0}^{q_h}\boldsymbol{\Omega}_k$, $\boldsymbol{\Omega}_C=[\boldsymbol{\Omega}_0\boldsymbol{1}_{\tilde{N}},\ldots,\boldsymbol{\Omega}_{q_H}\boldsymbol{1}_{\tilde{N}}]$, $\boldsymbol{\Omega}_D=\mathrm{diag}(\boldsymbol{1}_{\tilde{N}}^T\boldsymbol{\Omega}_0\boldsymbol{1}_{\tilde{N}},\ldots,\boldsymbol{1}_{\tilde{N}}^T\boldsymbol{\Omega}_{q_H}\boldsymbol{1}_{\tilde{N}})$ *and* $(\boldsymbol{\Omega}_k)_{ij}=K(\boldsymbol{u}_{f,N+(i-\tilde{N})+(k-q_H)},\boldsymbol{u}_{f,N+(j-\tilde{N})+(k-q_H)})$ *for* $i,j=1,\ldots,\tilde{N}$.

Proof. The proof is a straightforward extension of the proof of Lemma 1. From the conditions for optimality for $\boldsymbol{w}_k$ one obtains $\boldsymbol{w}_k=\boldsymbol{\Phi}_{(k)}(\boldsymbol{\alpha}+\boldsymbol{1}_{\tilde{N}}\beta_k)$. Substituting this along with $\boldsymbol{e}=\gamma^{-1}\boldsymbol{\alpha}$ into the constraints and replacing $\boldsymbol{\Phi}_{(k)}^T\boldsymbol{\Phi}_{(k)}$ by the kernel matrix $\boldsymbol{\Omega}_k$ then yields the dual problem. □

To evaluate the model at a new point one can use the one-step-ahead predictive equation

$$\hat{y}_t=\sum_{k=0}^{q_H}\boldsymbol{k}_k(\boldsymbol{u}_{f,t-k})^T(\boldsymbol{\alpha}+\boldsymbol{1}_{\tilde{N}}\beta_k)+d-\boldsymbol{a}_H^T\boldsymbol{y}_{t-1}, \tag{22}$$

where $\boldsymbol{k}_k(\boldsymbol{z})=[K(\boldsymbol{u}_{f,D+(k-q_H)},\boldsymbol{z}),\ldots,K(\boldsymbol{u}_{f,N+(k-q_H)},\boldsymbol{z})]^T\in\mathbb{R}^{\tilde{N}}$.

4.4. Recovery of the original model class

Problem (19) is guaranteed to have a rank-1 solution such that the original parameterization in terms of $\boldsymbol{w}$ and $\boldsymbol{b}_H$ can be obtained by computing a rank-1 factorization of $\boldsymbol{W}$. However, this is not the case for the relaxed problem (20). Therefore one has to resort to approximations. Recovering $\boldsymbol{w}$ from the matrix $\boldsymbol{W}$ is impossible if $\boldsymbol{\varphi}$ is only defined implicitly which is often the case. Therefore the emphasis is on finding a factor $\boldsymbol{b}_H$ that corresponds to $\boldsymbol{W}$. Then $\boldsymbol{b}_H$ can be fixed and $\boldsymbol{w}$ can be estimated using (18) as modeling constraint and the standard LS-SVM cost function $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+\frac{1}{2}\gamma\boldsymbol{e}^T\boldsymbol{e}$.

Let $\boldsymbol{W}$ have the singular value decomposition $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, then a possible approximation for $\boldsymbol{b}_H$ is given by the right singular vector corresponding to largest singular value. The matrix $\boldsymbol{V}$ containing the right singular values can be obtained from the eigenvalue decomposition of $\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T$. This matrix is finite dimensional and can be computed as

$$\boldsymbol{W}^T\boldsymbol{W}=\begin{bmatrix}\boldsymbol{\alpha}_0^T\boldsymbol{\Omega}_{0,0}\boldsymbol{\alpha}_0 & \cdots & \boldsymbol{\alpha}_0^T\boldsymbol{\Omega}_{0,q_H}\boldsymbol{\alpha}_{q_H}\\ \vdots & \ddots & \vdots\\ \boldsymbol{\alpha}_{q_H}^T\boldsymbol{\Omega}_{q_H,0}\boldsymbol{\alpha}_0 & \cdots & \boldsymbol{\alpha}_{q_H}^T\boldsymbol{\Omega}_{q_H,q_H}\boldsymbol{\alpha}_{q_H}\end{bmatrix}, \tag{23}$$

where $\boldsymbol{\alpha}_k=\boldsymbol{\alpha}+\boldsymbol{1}_{\tilde{N}}\beta_k$, $\boldsymbol{\Omega}_{k,l}=\boldsymbol{\Phi}_{(k)}^T\boldsymbol{\Phi}_{(l)}$ and $(\boldsymbol{\Omega}_{kl})_{ij}=K(\boldsymbol{u}_{f,N+(i-\tilde{N})+(k-q_H)},\boldsymbol{u}_{f,N+(j-\tilde{N})+(l-q_H)})$.

Using only the dominant singular value, the information in the remaining singular values is lost. Instead of solely relying on this information one can consider a randomized approach that is able to take all information into account. Therefore a vector $\boldsymbol{v}$ is drawn from a zero mean Gaussian distribution with covariance matrix $\boldsymbol{W}^T\boldsymbol{W}$. Then, one can generate estimates for $\boldsymbol{b}_H$ by choosing $\boldsymbol{b}_H=\boldsymbol{v}/\|\boldsymbol{v}\|_2$.

A third approach taken in Goethals et al. (2005, 2010, chap. 15) and Falck et al. (2009) is to notice that the predictions $\boldsymbol{w}_k^T\boldsymbol{\varphi}(\boldsymbol{u}_{f,t})$ should be linear dependent with the dependence given by $\boldsymbol{b}_H$. Therefore one can obtain another approximation for $\boldsymbol{b}_H$ from the singular value decomposition of the matrix of predictions $(\boldsymbol{Y}_T)_{tk}=\boldsymbol{w}_k^T\boldsymbol{\varphi}(\boldsymbol{u}_{f,t})=\boldsymbol{k}_k(\boldsymbol{u}_{f,t})^T(\boldsymbol{\alpha}+\boldsymbol{1}_{\tilde{N}}\beta_k)$ for $t=D,\ldots,N-q_H$.

Below a complete algorithm is given to estimate an over-parameterized LS-SVM Wiener–Hammerstein model and project it back onto the original model class.

Algorithm 4. Estimation of structured LS-SVM model.

1. Choose model orders p_H, q_H and q_f.
2. Select a regularization parameter γ and a kernel function K (and its parameters).
3. Compute the kernel matrices $\boldsymbol{\Omega}_k$, its average $\boldsymbol{\Omega}_A$, sums $\boldsymbol{\Omega}_D$ and row sums $\boldsymbol{\Omega}_C$ as defined in Lemma 2.
4. Solve the dual linear system (21).
5. Obtain estimate for $\boldsymbol{b}_H$ by one of the techniques described in Section 4.4.
6. Estimate a model with $\min_{\boldsymbol{w},c,\boldsymbol{a}_H,\boldsymbol{e}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+\frac{1}{2}\gamma\boldsymbol{e}^T\boldsymbol{e}$ subject to (18) for $t=D,\ldots,N$.

4.5. Handling large data sets

Standard LS-SVMs for NARX models can be extended to handle large data sets as described in Section 2. For the structured models discussed in this section a way to extend these ideas is outlined in the following. It has the disadvantage that it cannot handle as much data as in the unstructured case, but no modifications have to be made to the problem.

Using the approximate feature map $\hat{\boldsymbol{\varphi}}$ from Section 3 to form the matrices $\boldsymbol{\Phi}_{(k)}$ one can state the following corollary.

Corollary 3. *Let $\boldsymbol{\Phi}_C=[\boldsymbol{\Phi}_{(0)}^T,\ldots,\boldsymbol{\Phi}_{(q_H)}^T]^T$, and $\boldsymbol{\Phi}_S$ be a block matrix with $(\boldsymbol{\Phi}_S)_{kk}=\boldsymbol{\Phi}_{(k)}\mathbf{1}_{\tilde{N}}\in\mathbb{R}^{M\times 1}$ for $k=0,\ldots,q_H$ and all other blocks zero. Also define $\boldsymbol{S}=[\boldsymbol{\Phi}_C^T,-\boldsymbol{Y}^T,\mathbf{1}_{\tilde{N}}]$, $\boldsymbol{\omega}=[\boldsymbol{w}_0^T,\ldots,\boldsymbol{w}_{q_H}^T,\boldsymbol{a}_H^T,d]^T$ and $\boldsymbol{P}=[\boldsymbol{\Phi}_S^T,\mathbf{0}]^T$ where the block of zeros is of dimension $(q_H+1)\times(p_H+1)$. Then the extension of* (10) *to the structured model* (20) *is given by*

$$\begin{bmatrix}\boldsymbol{S}^T\boldsymbol{S}+\gamma^{-1}\boldsymbol{I}_{M,q_H,0} & \boldsymbol{P}\\ \boldsymbol{P}^T & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}\\ \boldsymbol{\beta}\end{bmatrix}=\begin{bmatrix}\boldsymbol{S}^T\boldsymbol{y}\\ \mathbf{0}\end{bmatrix}, \tag{24}$$

where $\boldsymbol{\beta}$ are the Lagrange multipliers for the centering constraints and $\boldsymbol{I}_{M,q_H,0}$ is a diagonal matrix of dimension $(M\cdot q_H+p_H+1)\times(M\cdot q_H+p_H+1)$ whose first $M\cdot q_H$ elements are one and all others are zero.

The one-step-ahead predictor for this overparameterized form is given by

$$\hat{y}_t=\mathrm{tr}(\boldsymbol{\Phi}_t^T\boldsymbol{W})+\boldsymbol{a}_H^T\boldsymbol{y}_{t-1}+d, \tag{25}$$

where $\boldsymbol{W}=[\boldsymbol{w}_0,\ldots,\boldsymbol{w}_{q_H}]$.

Whereas the unstructured model (3) can be solved from a linear system (10) in $M+1$ variables, the structured model (20) corresponds to a linear system (24) in $(M+1)\cdot q_H+p_H+1$ variables. Therefore even for a subsample with $M\ll\tilde{N}$, the solution can be computationally infeasible. The projection onto the original model class is straightforward as the matrix $\boldsymbol{W}$ is finite dimensional and explicitly known. Then the estimation procedure is a combination of Algorithms 3 and 4.

Algorithm 5. Estimation of structured fixed-size model.

1. Choose model orders p_H, q_H and q_f.
2. Select a subset $S_M\subset\{\boldsymbol{u}_{f,t}\}_{t=q_f+1}^{N}$ of $M\ll\tilde{N}$ data points from the data set.
3. Select a regularization parameter γ and a kernel function K (and its parameters).
4. Build the kernel matrix $\boldsymbol{\Omega}_M$ evaluated on S_M and compute its eigen decomposition $\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T$.
5. Use (7) to form $\boldsymbol{\Phi}_k$ by evaluating $\hat{\boldsymbol{\varphi}}$ for all $\boldsymbol{u}_{f,t}$.
6. Solve the primal linear system (24).
7. Obtain estimates for $\boldsymbol{w}$ and $\boldsymbol{b}_H$ from the dominant singular vectors of $\boldsymbol{W}$.
8. *Optional:* fix $\boldsymbol{b}_H$ and estimate a refined model with $\min_{\boldsymbol{w},c,\boldsymbol{a}_H,\boldsymbol{e}}\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+\frac{1}{2}\gamma\boldsymbol{e}^T\boldsymbol{e}$ subject to (18) for $t=D,\ldots,N$.

5. Evaluation on artificial data

This section provides empirical results based on an artificial case study relating the Wiener–Hammerstein overparameterization approach to the full black-box approach. In order to compare the different model structures on a Wiener–Hammerstein system under different conditions, several artificial data sets are generated. The first linear system $G(z)$ is given as a fourth order Chebychev type II digital low-pass filter with a stopband-edge frequency 0.5 (normalized) and a stopband ripple of 40 dB. The nonlinearity $f(\cdot)$ is a hyperbolic tangent, $\tanh(\cdot)$, representing a mildly nonlinear saturation characteristic. The second linear system $H(z)$ is given as a comb filter with five conjugate complex pairs of poles which are linearly spaced between 0.15 and 0.80; the filter has as many zeros as poles which are positioned at frequencies in the middle between consecutive poles (and one extra after the last pole). The transfer function is shown in Fig. 2. The output measurements are corrupted by zero mean white Gaussian noise which is filtered by the denominator of the system $H(z)$. The effect of the noise variance on the resulting performance is studied in the identification experiments, as well as the effect of the number of support vectors (SVs) is investigated. In each of these runs of experiments one parameter is varied over a set of values while the other parameter is kept fixed at its default value. The evaluated parameter values and their default values are shown in Table 1.

Numerical data sets are generated by driving the system with white Gaussian noise as excitation signal. For each of the combinations of parameter values that are investigated, a data set is generated. The first data points are discarded in order to eliminate transient effects. The remaining data is split up in $\tilde{N}\in\{100,250,500,1000,2500\}$ data points for model estimation (this number is determined by the number of support vectors that is used), 10,000 as validation set $\mathcal{V}_1$ to choose the regularization constant γ and the parameter of the Gaussian RBF kernel σ, 20,000 as validation set $\mathcal{V}_2$ to select model orders p,q and p_H,q_H,q_f, respectively, and finally 30,000 data points as independent test set.

Finally the compared model structures are

- NARX: LS-SVM model with NARX structure,
- STRCTRD: structured model,
- OVRPRZD: overparameterized model.

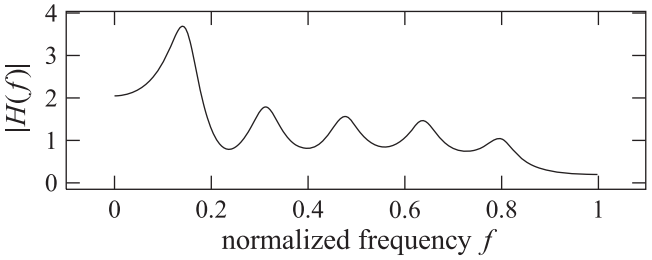


Fig. 2. The frequency response $H(z)$ of the comb filter used as the second linear block in the artificial Wiener–Hammerstein system data sets.

Table 1
Parameters used to generate artificial data sets. In a first set of experiments the noise variance is fixed at 0.01^2 while the number of support vectors is varied. Then in a second set of experiments the number of support vectors is set to 1000 while the different values for the noise variance are tested. The bold face numbers indicate the value which one parameter is fixed at, while the other one is varied.

Property	Values
Noise variance	$\mathbf{0.01}^2$, 0.10^2, 0.25^2, 0.35^2, 0.50^2
Number of SVs	100, 250, 500, **1000**, 2500

Table 2
List of considered model structures, the corresponding one-step-ahead predictors and the algorithms needed for their estimation.

Model	Model equation	Algorithm
NARX	(5)	1
STRCTRD	(18)[a]	4
OVRPRZD	(22)	4 (up to 4)
FS-NARX	(11)	3
FS-STRCTRD	(18)	5
FS-OVRPRZD	(25)	5 (up to 7)

[a] With $\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{u}_{f,t-k}) = \sum_{l=0}^{P} b_{H,l} \sum_{n=1}^{\hat{N}} \alpha_n K(\boldsymbol{u}_{f,t-k}, \boldsymbol{u}_{f,n-l})$.

The corresponding modeling equations and algorithms are summarized in Table 2.

5.1. Parameter tuning

For all experiments the Gaussian RBF kernel is used. Therefore a model is determined by the regularization parameter γ and the kernel bandwidth σ. Both values are selected by evaluating all their different combinations for 10 discretization of both parameters. For each combination the model is estimated and used to compute one-step-ahead predictions on the independent validation data $\mathcal{V}_1$. The values γ,σ that minimize the root mean squared error (RMSE) are then used in the following.

5.2. Model order selection

In case of the NARX model, the model order is selected with 1000 SVs and is then fixed and also used for all numbers of SVs. In case of the models with structural information, the model selection is performed based on OVRPRZD with 1000 SVs and then likewise used for all other structured models. To limit the number of models to be tested, the models orders p and q for the NARX model and p_H and q_H for the structured models are chosen identical, i.e., $p=q$ and $p_H=q_H$, respectively. Then, for each value of p or each combination p_H and q_f the optimal model parameters γ and the kernel parameter σ are estimated. To select the optimal model orders, the one-step-ahead predictions on the independent validation set $\mathcal{V}_2$ are computed. Finally the model order giving rise to the best RMSE is chosen.

5.3. Projection of the structured models

As detailed in Section 4.4, the model STRCTRD is the result of projecting the model OVRPRZD back onto the original model class. The quality of both models is determined by how close the solution $\boldsymbol{W}$ is to a rank-1 matrix. In case a rank-1 matrix would be obtained from (20) it would also be a solution of (19) as it satisfies the nonconvex rank constraint.

For the results presented here the deterministic projection method based on $\boldsymbol{W}$ is used. The quality of randomly drawn candidates is mostly too low, meaning that a large fraction of the energy of $\boldsymbol{W}$ is not concentrated in its dominant singular value. The performance of the two deterministic approaches is similar, but the approach working on $\boldsymbol{W}$ is slightly better than the procedure using predictions.

5.4. Influence of noise variance

The presence of noise strongly influences the quality of a model. Therefore the performance of the different model structures is compared for several levels of the noise variance. As the amount of noise is increased, the performance of all models deteriorates. For a fixed noise level, the one-step-ahead performance of the NARX model is better compared to the STRCTRD and OVRPRZD models, however the performance of iterative simulation is inferior for a noise variance of 0.25^2 or below. The performance has similar values for higher noise variances. The results are shown in Table 3. If the amount of measurement noise is low one can conclude from this experiment that the advantage of incorporating the structure of the system into the model will result in a significant increase of performance. If there is much noise, the full black-box NARX model performs equally well.

Table 3
RMSE $\times 10^3$ on the test data for artificial systems for different noise levels. The first value gives the performance of the model in recursive simulation mode; the value between parentheses is its one-step-ahead performance. The number of support vectors is fixed to 1000 SVs.

Noise var	NARX	STRCTRD	OVRPRZD
0.01^2	99.2 (2.3)	51.5 (28.3)	48.9 (29.2)
0.1^2	207.9 (20.8)	174.1 (123.3)	181.5 (130.6)
0.25^2	402.5 (86.0)	385.7 (270.4)	394.6 (282.1)
0.35^2	578.6 (173.0)	585.6 (407.3)	573.4 (409.6)
0.5^2	757.4 (290.2)	770.2 (535.9)	754.1 (532.1)

Table 4
RMSE $\times 10^3$ on the test data for artificial systems for different numbers of support vectors. The first value gives the performance of the model in recursive simulation mode; the value between parentheses is its one-step-ahead performance. The noise variance is fixed to 0.01^2.

# SVs	NARX	STRCTRD	OVRPRZD
100	132.09 (7.3)	127.9 (53.0)	164.8 (56.5)
250	135.72 (5.7)	94.0 (44.2)	129.7 (50.4)
500	155.51 (4.4)	62.3 (32.3)	85.9 (39.5)
1000	99.21 (2.3)	52.0 (28.6)	49.4 (29.5)
2500	50.29 (0.8)	49.8 (27.6)	40.3 (24.0)

5.5. Influence of number of data

This subsection considers the influence of the number of data available for estimating a model. As in the previous subsections all other parameters are kept at their nominal values as indicated in Table 1. As the number of data available for estimating the model increases, all models yield improved performance. Especially in terms of the performance on recursive simulation, the STRCTRD and OVRPRZD models perform significantly better than the NARX model as the number of estimation data points increases. The results are shown in Table 4.

Tests with higher model orders for $G(z)$ and different numbers of resonant peaks of $H(z)$ exhibit similar behavior.

5.6. Conclusions

In this section the effect of the noise and the number of data on the resulting model is studied. It turns out that exploiting the Wiener–Hammerstein structure in the model results in a substantial increase of performance if the amount of measurement noise in the data is low. Moreover, if the amount of available data is limited, the benefit of structured models over the full black-box NARX model is even more pronounced. As more data points are available, all models will result in improved performance, thereby respecting the above findings.

6. Results on the Wiener–Hammerstein benchmark data set

6.1. Description of data set

This section describes results of all developed methods applied on the Wiener–Hammerstein benchmark data set (Schoukens

et al., 2009). The Wiener–Hammerstein benchmark data set consists of input and output data $\{(u_t,y_t)\}_{t=1}^{188,000}$ which were obtained from measurements of a real-life electronic nonlinear system. The available data are obtained with a low noise level, and therefore one expects that exploiting the system structure will be advantageous. Moreover, the availability of a large number of measurements suggests the use of fixed-size and large-scale methods. In addition to NARX, STRCTRD and OVRPRZD the following fixed-size model structures are compared:

- FS-NARX: fixed-size LS-SVM model with NARX structure,
- FS-STRCTRD: fixed-size structured model,
- FS-OVRPRZD: fixed-size overparameterized model.

Model equations and algorithms for these fixed-size models are also given in Table 2. In order to compare their predictive powers, the number of support vectors is varied. In case of FS-NARX the number of data used for estimation is always $\tilde{N} \simeq 50{,}000$ and the number of support vectors corresponds to the dimension of the approximated feature map M. For all other models the number of support vectors is equal to the number of estimation data $\tilde{N}$.

The data set containing 188,000 measurements is split into several parts. The first 10,000 data points are discarded. The following 90,000 data points of the estimation data are then split up into blocks of 50,000 data points for model estimation, 20,000 data points as validation set $\mathcal{V}_1$ to select the regularization parameter γ and bandwidth σ of the Gaussian RBF kernel and 20,000 data points as additional validation set $\mathcal{V}_2$ to select the model orders p, q and p_H, q_H, q_f, respectively. The remaining 88,000 data points from the complete data set are left untouched during the whole model selection and estimation process. They are only used to assess the quality of the finally obtained models.

6.2. Model order selection

Model order selection and selection of the regularization parameter γ and the kernel bandwidth σ of the Gaussian RBF kernel are performed as outlined in the previous section. For the NARX models the result is shown in Fig. 3, the optimal model order is $p=q=16$. In case of the structured models the values $q_f=10$ and $p_H=q_H=14$ are obtained as shown in Fig. 4.

It should be noted that both the structured as well as the unstructured models achieve similar (good) performance over a wide range of model orders. As for the fixed-size structured models the computational cost scales with q_H, the values $p_H=q_H=12$ and $q_f=14$ are used, as this only results in a slight drop of the prediction performance.

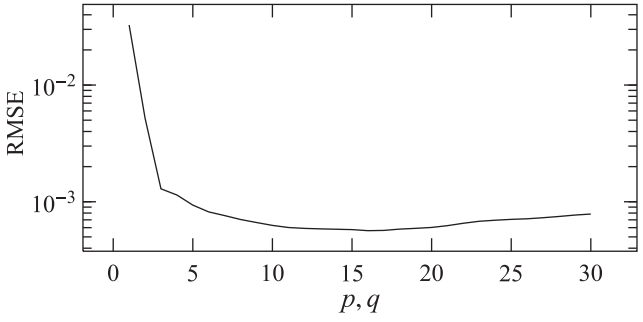


Fig. 3. Model order selection for NARX model for the Wiener–Hammerstein benchmark data set. The RMSE is computed for one-step-ahead predictions on the independent test set $\mathcal{V}_2$. The order of the NARX model is determined by the input and output orders p and q, respectively, representing the number of lags as in (12) and (13). The model orders p and q are chosen identical in the experiments, i.e., $p=q$.

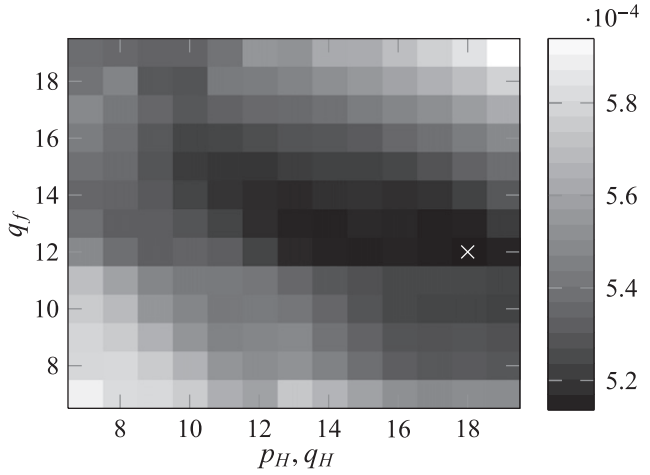


Fig. 4. Model selection for the structured Wiener–Hammerstein model. The RMSE for one-step-ahead prediction on the independent test set $\mathcal{V}_2$ is shown. The scalars p_H and q_H denote the number of poles and number of zeros of the model parameterization of the second linear block $H(z)$ as in (13). In the experiments the model orders p_H and q_H are chosen identical, i.e., $p_H=q_H$. The scalar q_f denotes the number of lagged inputs used to jointly model the first linear system $G(z)$ and the nonlinearity $f(\cdot)$ as a NFIR model. The little cross ($\times$) marks the optimal model order $p_H=q_H=18$ and $q_f=12$.

Table 5
RMSE $\times 10^3$ on test set for the Wiener–Hammerstein benchmark data set and different number of support vectors. The first value gives the performance of the model in recursive simulation mode; the value between parentheses is its one-step-ahead performance. The methods reported in this table have access to a number of data points equal to the number of support vectors (#SVs $=\tilde{N}$).

# SVs	NARX	STRCTRD	OVRPRZD
100	41.33 (5.54)	20.44 (0.74)	57.04 (0.86)
250	52.27 (1.59)	15.09 (0.67)	27.84 (0.66)
500	29.31 (0.75)	13.27 (0.61)	13.99 (0.57)
1000	12.75 (0.56)	13.17 (0.57)	10.94 (0.51)
2500	10.04 (0.52)	7.95 (0.50)	10.00 (0.48)
5000	9.50 (0.50)	6.22 (0.48)	6.06 (0.43)
10,000	8.89 (0.49)	5.84 (0.47)	4.60 (0.40)

Table 6
RMSE $\times 10^3$ for recursive simulation on test set for the Wiener–Hammerstein benchmark data set and different number of support vectors. The number of data used for estimation in the so-called fixed-size models is always $\tilde{N} \simeq 50{,}000$ and the number of support vectors corresponds to the dimension of the approximated map (#SVs $=M \ll \tilde{N}$). The estimate for b_H used for FS-STRCTRD with 2500 SVs is obtained from FS-OVRPRZD with 1000 SVs.

# SVs	FS-NARX	FS-STRCTRD	FS-OVRPRZD
100	23.46 (0.65)	8.74 (0.51)	7.00 (0.43)
250	9.38 (0.52)	5.65 (0.54)	4.51 (0.40)
500	8.85 (0.48)	4.46 (0.45)	3.90 (0.39)
1000	8.41 (0.47)	4.27 (0.44)	3.85 (0.38)
2500	5.08 (0.43)	4.18 (0.44)	–

Regarding the quality of approximation of the structured models, it is noted that, for the benchmark data, about 60% of the energy in $\boldsymbol{W}$ is concentrated in the largest eigenvalue.

6.3. Performance for different number of support vectors

The main results in terms of prediction performance for different number of support vectors are given in Tables 5 and 6. For the

recursively simulated values one can see that including structural information is able to increase the performance substantially over the NARX model. In case of the structured models STRCTRD is much better than OVRPRZD for small numbers of support vectors ($\tilde{N} < 500$). In contrast to all other models, the fixed-size models use 50,000 data points for the estimation. The results clearly indicate that in the presence of many measurements the prediction performance can be significantly improved. As with the previous models including model structure yields better models than the simple NARX structure.

Table 7 and 8 give the values requested by the benchmark definition (Schoukens et al., 2009) to enable an easy comparison with other methods. The values are the RMSE, the standard deviation and the mean of the residual signal on the test set and on the estimation set, respectively. All values are given for the best models shown in Tables 5 and 6. Time domain and frequency domain plots of the residual signal are given in Figs. 5 and 6.

Table 7

The root mean squared error $\times 10^3$ (RMSE), the standard deviation $\times 10^3$ (s) and the mean $\times 10^3$ (μ) of the prediction error $y_t - \hat{y}_t$. All values are shown both for the estimation data ($t =$ 1,001, …, 100,000, subscript E) and for the test data ($t =$ 101,001, …, 188,000, subscript T) of the Wiener–Hammerstein benchmark data set. All values are computed for recursive simulations of $\hat{y}_t$. Too ease the comparison all models are evaluated for 10,000 SVs.

	NARX	STRCTRD	OVRPRZD
RMSE_E	9.36	6.00	4.59
s_E	9.33	5.99	4.55
μ_E	−0.77	0.26	0.59
RMSE_T	8.88	5.84	4.60
s_T	8.88	5.84	4.56
μ_T	0.14	0.18	0.59

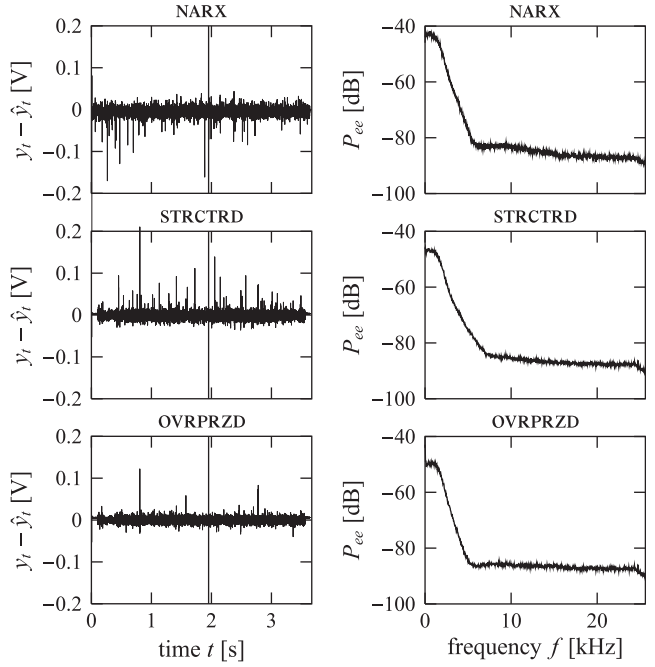


Fig. 5. Time and frequency domain plots for prediction error $y_t - \hat{y}_t$ for the NARX and structured models on the Wiener–Hammerstein benchmark data set. All values are computed for recursive simulations of $\hat{y}_t$. The best models from Table 5 are shown. The standard deviation of the original signal y_t is 0.2385 V.

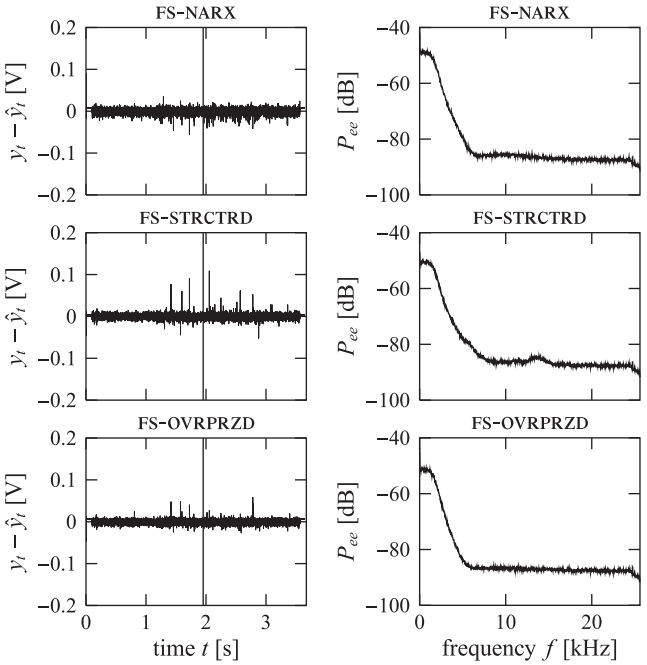


Fig. 6. Time and frequency domain plots for prediction error $y_t - \hat{y}_t$ for the fixed-size models on the Wiener–Hammerstein benchmark data set. All values are computed for recursive simulations of $\hat{y}_t$. The best models from Table 6 are shown. The standard deviation of the original signal y_t is 0.2385 V.

Table 8

The root mean squared error $\times 10^3$ (RMSE), the standard deviation $\times 10^3$ (s) and the mean $\times 10^3$ (μ) of the prediction error $y_t - \hat{y}_t$. All values are shown both for the estimation data ($t = 1001, \ldots, 100,000$, subscript E) and for the test data ($t =$ 101,001, …, 188,000, subscript T) of the Wiener-Hammerstein benchmark data set. All values are computed for recursive simulations of $\hat{y}_t$. Note that the models summarized in this table have access to $\tilde{N} \simeq 50,000$ estimation data points as opposed to the methods from Table 7 in which the number of available estimation data points corresponds to the number of support vectors. The results are shown for the best models obtained in Table 6.

	FS-NARX	FS-STRCTRD	FS-OVRPRZD
RMSE_E	5.08	3.88	3.72
s_E	5.07	3.88	3.71
μ_E	0.22	0.09	0.22
RMSE_T	4.76	4.18	3.85
s_T	4.76	4.18	3.84
μ_T	0.25	0.06	0.23

7. Conclusions

This paper described the application of Least-Squares Support Vector Machines-based models to the Wiener–Hammerstein system identification problem. A review of basic LS-SVMs was given, together with a method to handle large-scale data sets. Models which incorporate information about the Wiener–Hammerstein structure were developed. The proposed models have been tested on a series of artificially created Wiener–Hammerstein systems in which the amount of noise and the amount of measurements were varied. One can conclude that all considered model structures benefit from an increased number of support vectors. The performance of the structured models was in general better than for the NARX model and converges faster as the number of support vectors increases. In the presence of larger noise variance the advantage of the structural information shrank. For one-step-ahead predictions the NARX model were even better.

The results of this analysis were validated on the identification of the Wiener–Hammerstein benchmark problem. The experiments gave evidence to the same conclusions as obtained in the artificial examples. All models converged quite fast as the number of support vectors grew. Due to the large size of the data set, tests with fixed-size models were performed. These were able to make use of the whole data set and lead to improved prediction performances when compared to their classical counterparts. As before the structured models performed better than the NARX model. Finally the resulting models have a performance which is among the best black-box models approaches presented in the SYSID 2009 invited session, suggesting that the presented framework serves as a convenient method for Wiener–Hammerstein system identification.

Acknowledgments

Research supported by Research Council KUL: GOA/10/09 MaNet, PFV/10/002 (OPTEC), several PhD/postdoc and fellow grants; Flemish Government: IOF: IOF/KP/SCORES4CHEM; FWO: PhD/postdoc grants, projects: G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain–machine), G.0377.09 (Mechatronics MPC), G.0377.12 (Structured systems) research community (WOG: MLDM); IWT: PhD grants, projects: Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&ODsquare. Belgian Federal Science Policy Office: IUAP P7/ (DYSCO, Dynamical systems, control and optimization, 2012–2017); IBBT; EU: ERNSI, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B; COST: Action ICO806: IntelliCIS; Contract Research: AMINAL; Other: ACCM. P.D. is a research assistant at the KU Leuven, Belgium and is supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). K.P. is an assistant professor at UU/IT, supported in part by the Swedish Research Council under contract 621-2007-6364. B.D.M. is a full professor at the KU Leuven, Belgium. J.S. is a professor at the KU Leuven, Belgium. The scientific responsibility is assumed by its authors.

References

Bai, E.-W. (1998). An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems. *Automatica*, *34*, 333–338.

Bai, E.-W., & Liu, Y. (2007). Recursive direct weight optimization in nonlinear system identification: A minimal probability approach. *IEEE Transactions on Automatic Control*, *52*, 1218–1231.

Bershad, N. J., Celka, P., & Mclaughlin, S. (2001). Analysis of stochastic gradient identification of Wiener–Hammerstein systems for nonlinearities with Hermite polynomial expansions. *IEEE Transactions on Signal Processing*, *49*, 1060–1072.

Billings, S. A., & Fakhouri, S. Y. (1978). Identification of a class of nonlinear systems using correlation analysis. *Proceedings of IEE*, *125*, 691–697.

Boutayeb, M., & Darouach, M. (1995). Recursive identification method for MISO Wiener–Hammerstein model. *IEEE Transactions on Automatic Control*, *40*, 287–291.

Boyd, S. P., & Chua, L. (1983). Uniqueness of a basic nonlinear structure. *IEEE Transactions on Circuits and Systems*, *30*, 648–651.

Boyd, S. P., & Chua, L. (1985). Uniqueness of circuits and systems containing one nonlinearity. *IEEE Transactions on Automatic Control*, *30*, 674–681.

Chang, F. H. I., & Luus, R. (1971). A noniterative method for identification using Hammerstein model. *IEEE Transactions on Automatic Control*, *16*, 464–468.

De Brabanter, K., De Brabanter, J., Suykens, J. A. K., & De Moor, B. (2010). Optimized fixed-size kernel models for large data sets. *Computational Statistics and Data Analysis*, *54*, 1484–1504.

De Brabanter, K., Dreesen, P., Karsmakers, P., Pelckmans, K., De Brabanter, J., Suykens, J. A. K., et al. (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). Saint-Malo, France.

De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, et al. (2010). *LS-SVMlab toolbox user's guide version 1.8*. Internal report 10-146, ESAT-SISTA. Leuven, Belgium: KU Leuven.

Enqvist, M., & Ljung, L. (2005). Linear approximations of nonlinear FIR systems for separable input processes. *Automatica*, *41*, 459–473.

Espinoza, M., Suykens, J. A. K., Belmans, R., & De Moor, B. (2007). Electric Load Forecasting—Using kernel based modeling for nonlinear system identification. *IEEE Control Systems Magazine*, *27*, 43–57.

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). Saint-Malo, France.

Falck, T., Signoretto, M., Suykens, J. A. K., & De Moor, B. (2011). *A two stage algorithm for kernel based partially linear modeling with orthogonality constraints*. Internal report 10-03, ESAT-SISTA. Leuven, Belgium: KU Leuven.

Giri, F., & Bai, E.-W. (Eds.). (2010). *Block-oriented nonlinear system identification. Lecture notes in control and information sciences* (Vol. 404). Springer.

Girolami, M. (2002). Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, *14*, 669–688.

Goethals, I., Pelckmans, K., Falck, T., Suykens, J. A. K., & De Moor, B. (2010). *NARX identification of Hammerstein systems using Least-Squares Support Vector Machines. Lecture notes in control and information sciences* (Vol. 404, pp. 241–256). Springer.

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*, 1263–1272.

Goodman, J., Herman, M., Bond, B., & Miller, B. (2009). A log-frequency approach to the identification of the Wiener–Hammerstein model. *IEEE Signal Processing Letters*, *16*, 889–892.

Greblicki, W., & Pawlak, M. (2008). *Non-parametric system identification*. Cambridge University Press.

Kibangou, A. Y., & Favier, G. (2006). Wiener-Hammerstein systems modeling using diagonal Volterra kernels coefficients. *IEEE Signal Processing Letters*, *13*, 381–384.

Leontaritis, I. J., & Billings, S. A. (1985). Input–output parametric models for non-linear systems. Part I: Deterministic non-linear systems. *International Journal of Control*, *41*, 303–328.

Lin, T., Horne, B., Tino, P., & Giles, C. (1996). Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, *7*, 1329–1338.

Lind, I., & Ljung, L. (2008). Regressor and structure selection in NARX models using a structured ANOVA approach. *Automatica*, *44*, 383–395.

Pillonetto, G., Quang, M. H., & Chiuso, A. (2011). A new kernel-based approach for nonlinear system identification. *IEEE Transactions on Automatic Control*, *56*, 2825–2840.

Piroddi, L., & Spinelli, W. (2003). An identification algorithm for polynomial NARX models based on simulation error minimization. *International Journal of Control*, *76*, 1767–1781.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Springer.

Roll, J., Nazin, A., & Ljung, L. (2005). Nonlinear system identification via direct weight optimization. *Automatica*, *41*, 475–490.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Schoukens, J., Suykens, J. A. K., & Ljung, L. (2009). Wiener–Hammerstein benchmark. In *Proceedings of the 15th IFAC symposium on system identification* (*SYSID 2009*). Saint-Malo, France.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., et al. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, *31*, 1691–1724.

Spinelli, W., Piroddi, L., & Lovera, M. (2005). On the role of prefiltering in nonlinear system identification. *IEEE Transactions on Automatic Control*, *50*, 1597–1602.

Suykens, J. A. K., Alzate, C., & Pelckmans, K. (2010). Primal and dual model representations in kernel-based learning. *Statistics Surveys*, *4*, 148–183.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least Squares Support Vector Machines*. World Scientific.

Tan, A. H., & Godfrey, K. (2002). Identification of Wiener–Hammerstein models using linear interpolation in the frequency domain (LIFRED). *IEEE Transactions on Instrumentation and Measurement*, *51*, 509–521.

Vapnik, V. N. (1998). *Statistical learning theory*. John Wiley and Sons.

Wahba, G. (1990). *Spline models for observational data*. SIAM.

Williams, C. K. I., & Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In: T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, Vol. 13 (pp. 682–688). MIT Press.