Preprints of the
15th IFAC Symposium on System Identification
Saint-Malo, France, July 6-8, 2009

Identification of Wiener-Hammerstein Systems using LS-SVMs

Tillmann Falck *, **Kristiaan Pelckmans** *,
Johan A.K. Suykens *, **Bart De Moor** *

* *Katholieke Universiteit Leuven - ESAT - SCD/SISTA,*
Kasteelpark Arenberg 10,
B-3001 Leuven (Heverlee), Belgium
email: {tillmann.falck,kristiaan.pelckmans,johan.suykens,bart.demoor}
@esat.kuleuven.be

Abstract: This paper extends the overparametrization technique as used for Hammerstein systems employing nonlinear Least-Squares Support Vector Machines (LS-SVMs) towards the identification of Wiener-Hammerstein systems. We present some practical guidelines as well as empirical results on the performance of the method with respect to various deficiencies of the excitation signal. Finally we apply our method to the SYSID2009 Wiener-Hammerstein benchmark data set.

Keywords: Wiener-Hammerstein systems, Identification, Over-parametrization.

1. INTRODUCTION

Wiener-Hammerstein systems [Billings and Fakhouri, 1978] are an important special class of nonlinear systems consisting of a concatenation of two linear subsystems with a static nonlinearity in between them as shown in Figure 1. There exists extensive literature on the estimation of the easier Hammerstein systems, see for example Narendra and Gallman [1966], Pawlak [1991] or Bai and Fu [2002]. Yet for the case of Wiener-Hammerstein systems all known methods are limited to rather simple models restricting attention to the case of low order linear dynamic blocks and simple nonlinearities, see e.g. Tan and Godfrey [2002], Bershad et al. [2001], Enqvist and Ljung [2005] and Greblicki and Pawlak [2008]. One popular approach for identification of Hammerstein systems known in literature is overparametrization, dividing the identification in two stages [Chang and Luus, 1971, Bai, 1998]: identification of a slightly broader model class, and projection of the estimate on the strict Hammerstein class. This paper extends this technique to handle input dynamics.

The proposed algorithm extends the result in Goethals et al. [2005] where the static nonlinearity is modeled by Least Squares - Support Vector Machines (or LS-SVMs). LS-SVMs [Suykens et al., 2002] are a variation of the classical Support Vector Machines (SVMs) [Vapnik, 1998, Schölkopf and Smola, 2002], but considering equality instead of inequality constraints and using a L_2 loss. In this way the methodology can be extended to a wider class of problems in supervised and unsupervised learning. The technique is closely related to the technique of smoothing splines [Wahba, 1990]. Support vector machines perform nonlinear regression by projecting the input into a high (possibly infinite) dimensional space and then solving a regularized linear regression problem in that feature space. The mapping into the feature space itself is typically not known explicitly, but is induced by a proper positive definite kernel function. A main advantage of doing so is that the dual problem corresponding to the original (primal) optimization (estimation) problem can be expressed in terms of a finite number of unknowns, and allows for efficient solvers to be used. An important difference between SVMs and LS-SVMs is that the latter can be solved as a linear system, while the former needs to solve a convex quadratical program.

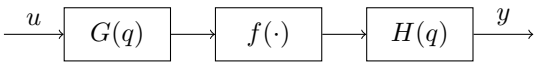


Fig. 1. General structure of a Wiener-Hammerstein system

The contribution of this paper is to show that the combination of overparametrization with LS-SVMs as in Goethals et al. [2005] yields a powerful algorithm to identify Wiener-Hammerstein systems. The key idea is to jointly model the input linear dynamic block with the nonlinearity, and apply an overparametrization technique to recover the parameters of the linear output system $H(q)$. The novelty of this approach is firstly that it assumes no prior knowledge of any of the blocks and secondly, it does not rely on inversion at any stage. Thus the class of static nonlinearities is only restricted by the properties of the used kernel function (which often implies smoothness of the estimate) and especially there are no restrictions on the zeros and poles of the second linear system. The only assumption is that $G(q)$ has finite memory. From a practical perspective the model order of $G(q)$ should not be too high. This can be seen as follows: if the expressive power of the function covering $H(q)$ and f would get too large, one would be able to capture the output dynamics with this model, and the output parameters cannot be estimated accurately. In general, as we consider the identification of the first linear block and the nonlinearity jointly, the technique of regularization becomes even more crucial to avoid (numerical) ill-conditioning than in the simpler

Hammerstein identification methods. We investigate and compare the use of the proposed estimation scheme with respect to prediction performance for cases where the input signal is suboptimal (colored, short), and would result in bad estimates in the purely linear case [Söderström and Stoica, 1989]. Furthermore we report the performance of the proposed method on the SYSID2009 benchmark data Schoukens et al. [2008].

The paper uses the following notation. Vector valued variables are written as lowercase, boldface letters whereas matrices are denoted by uppercase, boldface characters. All variables that are not boldfaced are scalar valued. Constants are usually put as capital letters. The backshift operator q is defined as $qx_t = x_{t-1}$. With abuse of notation the subscript of a variable might refer to both the element of a vector or the element of a sequence. Functions, denoted by lowercase names, are real valued and always have real domains. Systems are specified by their transfer functions which are in q and typeset in uppercase letters. Estimated values are highlighted with a hat.

The paper is organized as follows. Section 2 describes the proposed method, whereas in Section 3 it is extended to large data sets. In Section 4 the method is applied to numerical data in particular it is evaluated on the benchmark data. The next section is concerned with the analysis with respect to different excitation signal properties. Conclusions are given in Section 6.

2. IDENTIFICATION OF WIENER-HAMMERSTEIN SYSTEMS

The technique presented in Goethals et al. [2005] is extended towards the convex identification of another nonlinear block oriented model, defined as follows.

Definition 1. (Wiener-Hammerstein System). The system shown in Figure 1 consisting of a cascade of a linear dynamic system, static nonlinearity and finally a dynamic linear system is called Wiener-Hammerstein system. It is described by

$$y_t = H(q)(f(G(q)u_t)) \tag{1}$$

where f typically possesses a smoothness property.

Let $G(q)$ and $H(q)$ be parameterized as a FIR and an ARX model respectively. The FIR part has order $P \in \mathbb{N}$ while the ARX part has orders $R_1, R_2 \in \mathbb{N}$ respectively with parameters $\boldsymbol{a} = [a_0, \ldots, a_{P-1}]^T \in \mathbb{R}^P$, $\boldsymbol{b} = [b_0, \ldots, b_{R_1-1}]^T$ and $\boldsymbol{c} = [c_0, \ldots, c_{R_2-1}]^T$.

Given a set of measurements $\{(u_t, y_t)\}_{t=1}^T$ of inputs and outputs the proposed method estimates the parameters $\boldsymbol{b}$ and $\boldsymbol{c}$ for $H(q)$ and a dynamical nonlinear block describing $f(G(q)u_t)$. For notational ease define vectors of lagged inputs $\boldsymbol{u}_t = [u_t, \ldots, u_{t-D+1}]^T$ and outputs $\boldsymbol{y}_t = [y_{t-1}, \ldots, y_{t-R_2}]^T$. Then the full model is given by

$$\hat{y}_t = \sum_{r=0}^{R_1-1} b_r f(\boldsymbol{a}^T \boldsymbol{u}_{t-r}) + \boldsymbol{c}^T \boldsymbol{y}_t. \tag{2}$$

For further clarity let the index τ be defined as $\tau = \max(P + R_1, R_2)$.

The identification of a Wiener-Hammerstein system based on observations can be formalized as

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},f} \mathcal{J}(f, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \\ &= \min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},f} \sum_{t=\tau}^{T} \left(y_t - \sum_{r=0}^{R_1-1} b_r f(\boldsymbol{a}^T \boldsymbol{u}_{t-r}) + \boldsymbol{c}^T \boldsymbol{y}_t \right)^2. \end{aligned} \tag{3}$$

Definition 2. (Overparametrisation). The method of overparametrization starts by representing the Wiener-Hammerstein model as

$$y_t = \sum_{r=0}^{R_1-1} f_r(\boldsymbol{u}_{t-r}) + \boldsymbol{c}^T \boldsymbol{y}_t + d + e_t \tag{4}$$

where we implicitly assume that $f_r(\boldsymbol{x}) = b_r f(\boldsymbol{a}^T \boldsymbol{x})$ for all $r = 0, \ldots, R_1 - 1$. Without loss of generality we choose that the individual functions f_r are zero mean. After identifying the unknowns $\{f_r\}$ and $\{c_r\}$, the obtained model is projected back onto the Wiener-Hammerstein class. In particular, we decompose the functions $\{f_r\}$ into a global f and linear parameters $\{b_r\}$ by computing the best rank one approximation.

2.1 Identification

We consider the identification using kernel based models using the LS-SVM formalism [Suykens et al., 2002], as illustrated in Goethals et al. [2005]:

$$\min_{\boldsymbol{w}_r,\boldsymbol{c},d,e_t} \frac{1}{2} \sum_{r=0}^{R_1-1} \boldsymbol{w}_r^T \boldsymbol{w}_r + \frac{\gamma}{2} \sum_{t=\tau}^{T} e_t^2 \tag{5a}$$

subject to

$$y_t = \sum_{r=0}^{R_1-1} \boldsymbol{w}_r^T \boldsymbol{\varphi}(\boldsymbol{u}_{t-r}) + \boldsymbol{c}^T \boldsymbol{y}_t + d + e_t \quad \forall t = \tau, \ldots, T \tag{5b}$$

$$\sum_{t=\tau}^{T} \boldsymbol{w}_r^T \boldsymbol{\varphi}(\boldsymbol{u}_{t-r}) = 0 \ \forall r = 0, \ldots, R_1 - 1 \tag{5c}$$

where the $\boldsymbol{w}_r$ parameterize the nonlinear functions f_r, $\boldsymbol{\varphi} : \mathbb{R}^D \to \mathbb{R}^{n_h}$ is the feature map, $d \in \mathbb{R}$ denotes the offset and e_t is the sequence of estimation residuals. The model (4) is incorporated as (5b). The second constraint (5c) is used to center the nonlinear functions and hence ensures that the estimated functions are empirically zero mean. This is helpful for the final projection.

Define a data matrix $\boldsymbol{Y} = [\boldsymbol{y}_\tau, \ldots, \boldsymbol{y}_T]^T$ for the linear AR part of the model and a vector of outputs $\boldsymbol{y} = [y_\tau, \ldots, y_T]^T$. Let $\boldsymbol{1} \in \mathbb{R}^{T-\tau}$ denote a vector of all ones. After forming the Lagrangian for (5) and solving the conditions for optimality, the solution to the estimation problem can be found by solving the linear system

$$\begin{bmatrix} (\boldsymbol{\Omega}_A + \gamma_1^{-1}\boldsymbol{I}) & \boldsymbol{\Omega}_C & \boldsymbol{1} & \boldsymbol{Y} \\ \boldsymbol{\Omega}_C^T & & & \diagup \\ \boldsymbol{1}^T & & 0 & \\ \boldsymbol{Y}^T & \diagup & & \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ d \\ \boldsymbol{c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0} \\ 0 \\ \boldsymbol{0} \end{bmatrix}. \tag{6}$$

Let $\boldsymbol{\Omega}^{r,s} \in \mathbb{R}^{(T-\tau)\times(T-\tau)}$ be shifted kernel matrices defined as

$$\Omega_{i,j}^{r,s} = K(\boldsymbol{u}_{i-r}, \boldsymbol{u}_{j-s}) = \boldsymbol{\varphi}(\boldsymbol{u}_{i-r})^T \boldsymbol{\varphi}(\boldsymbol{u}_{j-s}) \tag{7}$$

for all $i, j = \tau, \ldots, T$. The summed kernel matrix $\boldsymbol{\Omega}_A$ can be written as

$$\boldsymbol{\Omega}_A = \sum_{r=0}^{R_1-1} \boldsymbol{\Omega}^{r,r} \tag{8}$$

whereas the R_1 centering constraints are represented by

$$\boldsymbol{\omega}_C^r = \boldsymbol{\Omega}^r \mathbf{1} \tag{9}$$

and collected in $\boldsymbol{\Omega}_C = [\boldsymbol{\omega}_C^0, \ldots, \boldsymbol{\omega}_C^{R_1-1}]$.

The individual estimated functions can be evaluated with

$$\hat{f}_r(\boldsymbol{x}) = \sum_{k=\tau}^{T} (\alpha_k + \beta_r) K(\boldsymbol{x}, \boldsymbol{u}_{k-r}) \tag{10}$$

and the global overparameterized model can be evaluated as

$$\hat{f}(\boldsymbol{u}_t, \ldots, \boldsymbol{u}_{t-R_1+1}, \boldsymbol{y}_t) = \sum_{r=0}^{R_1-1} \hat{f}_r(\boldsymbol{u}_{t-r}) + \boldsymbol{c}^T \boldsymbol{y}_t + d. \tag{11}$$

2.2 Projection by Rank-one Approximation

The second stage of the overparametrization approach projects the R_1 identified nonlinear functions onto the set of linear parameters $\{\hat{b}_r\}_{r=0}^{R_1-1}$ and a set of nonlinear function values $\{\hat{f}(\boldsymbol{u}_t)\}_{t=2\tau}^{T}$. Therefore we form a matrix $\boldsymbol{F}$ that contains all values for the individual functions $\hat{f}_r(\cdot)$ evaluated o n the training data $\{\boldsymbol{u}_t\}_{t=2\tau}^{T}$ as in

$$\begin{bmatrix} * \\ \hline \boldsymbol{F} \\ \hline * \end{bmatrix} := \left[\begin{array}{cccc} \hat{f}_0(\boldsymbol{u}_\tau) & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ \hat{f}_0(\boldsymbol{u}_{2\tau-1}) & & & 0 \\ \hline \hat{f}_0(\boldsymbol{u}_{2\tau}) & & & \hat{f}_{R_1-1}(\boldsymbol{u}_{2\tau}) \\ \vdots & & & \vdots \\ \hat{f}_0(\boldsymbol{u}_T) & & & \hat{f}_{R_1-1}(\boldsymbol{u}_T) \\ \hline 0 & & & \hat{f}_{R_1-1}(\boldsymbol{u}_{T+1}) \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 0 & \hat{f}_{R_1-1}(\boldsymbol{u}_{T+\tau}) \end{array}\right] \tag{12}$$

where the blocks of zeros are lower and upper diagonal respectively and the matrix $\boldsymbol{F}$ is of dimension $(T-2\tau)\times R_1$.

Then the projection can be performed by taking the best rank one approximation of $\boldsymbol{F}$ as follows

$$\boldsymbol{F} = \begin{bmatrix} \hat{f}(\boldsymbol{u}_{2\tau}) \\ \hat{f}(\boldsymbol{u}_{2\tau+1}) \\ \vdots \\ \hat{f}(\boldsymbol{u}_T) \end{bmatrix} \begin{bmatrix} b_0 & \hat{b}_1 & \ldots & \hat{b}_{R_1-1} \end{bmatrix}. \tag{13}$$

This can be computed efficiently by computing the Singular Value Decomposition (SVD), and withholding the left and right singular vectors corresponding to the largest singular value.

2.3 Estimation of the Final Model

With the estimated values for $\{b_r\}_{r=0}^{R_1-1}$, the overparametrization and the centering constraints can be removed from the estimation problem in (5) yielding

$$\min_{\boldsymbol{w},\boldsymbol{c},d,e_t} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{t=\tau}^{T} e_t^2 \tag{14a}$$

subject to

$$y_t = \sum_{r=0}^{R_1-1} \hat{b}_r \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{u}_{t-r}) + \boldsymbol{c}^T \boldsymbol{y}_t + d + e_t \quad \forall t = \tau, \ldots, T \tag{14b}$$

The solution is again obtained by a solving a linear system

$$\begin{bmatrix} (\boldsymbol{\Omega}_B + \gamma_2^{-1}\boldsymbol{I}) & \mathbf{1} & \boldsymbol{Y} \\ \mathbf{1}^T & 0 & \mathbf{0} \\ \boldsymbol{Y}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ d \\ \boldsymbol{c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \\ \mathbf{0} \end{bmatrix} \tag{15}$$

with

$$\boldsymbol{\Omega}_B = \sum_{r=0}^{R_1-1}\sum_{s=0}^{R_1-1} \hat{b}_r \hat{b}_s \boldsymbol{\Omega}^{r,s}. \tag{16}$$

Estimates of the function are then obtained by

$$\hat{f}(\boldsymbol{u}_t, \ldots, \boldsymbol{u}_{t-R_1+1}, \boldsymbol{y}_t) = \sum_{k=\tau}^{T} \alpha_k \sum_{r=0}^{R_1-1}\sum_{s=0}^{R_1-1} K(\boldsymbol{u}_{t-r}, \tilde{\boldsymbol{u}}_{k-s}) + \boldsymbol{c}^T \boldsymbol{y}_t + d \tag{17}$$

where $\{\tilde{\boldsymbol{u}}_k\}_{k=\tau}^{T}$ is the set of inputs used during the estimation phase.

The complete approach in summarized in Algorithm 1.

Algorithm 1 Identification of Wiener-Hammerstein systems

(1) Compute the shifted kernel matrices $\boldsymbol{\Omega}^{r,s}$ in Eq. (7)
(2) Construct the summed kernel matrix $\boldsymbol{\Omega}_A$ and the centering constraints $\boldsymbol{\omega}_C^r$ in Eqs. (8) & (9)
(3) Solve the dual system for the component-wise LS-SVM in (6)
(4) Construct the matrix of predictions $\boldsymbol{F}$ in (12)
(5) Compute the SVD of $\boldsymbol{F}$ and take the rank one approximation as in Eq. (13)
(6) Compute a weighted kernel matrix $\boldsymbol{\Omega}_B$ as in Eq. (16)
(7) Solve the dual for the weighted LS-SVM in Eq. (15)

3. APPLICABILITY TO LARGE DATA SETS

In kernel based techniques the Gram matrix (7) is computed for all pairs of inputs. For large data sets this becomes infeasible as the storage requirement grows with the square of the training set size. Therefore approximation techniques have been developed to effectively process large amounts of data. For LS-SVMs the Nyström approximation [Williams and Seeger, 2001] of the feature map $\boldsymbol{\varphi}$ has been shown to be effective [De Brabanter et al., 2009]. In Fixed Size LS-SVMs [Suykens et al., 2002] the feature map is approximated for a subset of the training data based on the quadratic Renyi entropy, the explicit knowledge of the feature map then allows the solution of the estimation problem (5) and (14) in the primal domain using all data. For (14) the linear system is identical to the one in De Brabanter et al. [2009] except for a reweighting of the feature map according to the linear parameters b_r. The linear system for (5) is slightly more complicated as it is larger and includes the centering constraints. Let

Table 1. Estimated mean, standard deviation, root mean squared error and fit for the benchmark data. The model uses 1000 SVs and is trained using the whole estimation data.

	$\hat{\mu}$	$\hat{\sigma}$	RMSE	fit
estimation	$5.64 \cdot 10^{-5}$	$4.14 \cdot 10^{-3}$	$4.14 \cdot 10^{-3}$	98.3%
validation	$1.41 \cdot 10^{-4}$	$4.07 \cdot 10^{-3}$	$4.07 \cdot 10^{-3}$	98.3%

$\boldsymbol{\Phi}_r = [\hat{\boldsymbol{\varphi}}(\boldsymbol{u}_{\tau-R_1+1-r}), \ldots, \hat{\boldsymbol{\varphi}}(\boldsymbol{u}_{T-r})]$ for $r = 0, \ldots, R_1 - 1$ where $\hat{\boldsymbol{\varphi}} : \mathbb{R}^D \to \mathbb{R}^M$ is an approximation of the feature map based on M data points. Then define $\boldsymbol{D} = [\boldsymbol{\Phi}_0^T, \ldots, \boldsymbol{\Phi}_{R_1-1}^T, \boldsymbol{Y}, \boldsymbol{1}]$, $\boldsymbol{\Gamma}_I = \gamma^{-1} \begin{bmatrix} \boldsymbol{I}_{(R_1 \cdot M)^2} & \\ & \boldsymbol{0}_{(R_2+1)^2} \end{bmatrix}$ and a $R_1 \times R_1$ block matrix $\boldsymbol{\Phi}_C$ with blocks $(\boldsymbol{\Phi}_C)_{rr} = \boldsymbol{\Phi}_r \boldsymbol{1}$ of size $M \times 1$. Then the solution of (5) is obtained by solving

$$\left[\begin{array}{c|c} \boldsymbol{P}^T\boldsymbol{P} + \boldsymbol{\Gamma}_I & \boldsymbol{\Phi}_C \\ \hline \boldsymbol{\Phi}_C^T \ \boldsymbol{0}_{((R_2+1)\times R_1)} & \boldsymbol{0}_{(R_1\times(R_2+1))} \\ & \boldsymbol{0}_{(R_1)^2} \end{array}\right] \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^T\boldsymbol{y} \\ \boldsymbol{0}_{(R_1)} \end{bmatrix} \quad (18)$$

with $\boldsymbol{\theta} = [\boldsymbol{w}_0^T, \ldots, \boldsymbol{w}_{R_1-1}^T, \boldsymbol{c}^T, d]^T$ and the Lagrange multipliers $\boldsymbol{\beta} = [\beta_0, \ldots, \beta_{R_1-1}]^T$ corresponding to the centering constraints (5c).

4. EMPIRICAL ASSESSMENTS

4.1 Training Procedure

Especially in nonlinear estimation techniques the model selection is critical to the model performance on independent data. In the proposed method we have chosen to parameterize the model in terms of the bandwidth σ of the used RBF kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/\sigma^2)$ and the regularization parameters γ_1 and γ_2 for the systems in (6) and (15) respectively. Additionally the model orders R_1, R_2 and D have to be determined.

The selection is made in two stages. At first, we tune the model parameters $(\gamma_1, \gamma_2, \sigma)$ in Algorithm 1 such that the cross-validation performance becomes maximal. We found empirically that σ_1 and γ_1 should be chosen from a grid of possible parameters, while γ_2 in Eq. (16) can be tuned independently once γ_1 and σ_1 are fixed.

The outer level is used for model order selection. For all possible model orders (P, R_1, R_2), the above paragraph describes a method to tune $(\gamma_1, \gamma_2, \sigma_1)$. For application on the Benchmark data in the next subsection we found empirically that it is plausible to take $D \equiv R_1$. As a consequence, we only have to select proper values of R_1 and R_2 of a grid of values. For each possible pair of orders (R_1, R_2) on the grid, the performance is evaluated on an independent validation set.

4.2 Performance on the SYSID2009 Benchmark Data

The benchmark data set contains 188,000 samples of which the first 100,000 are to be used for training and model selection (model estimation) whereas the last 88,000 are kept untouched for the final performance evaluation (model validation). The modeling goal is simulation performance. To ensure stability the model selection is based directly on simulation performance (instead of just one-step ahead predictions). Due to the large number of the estimation

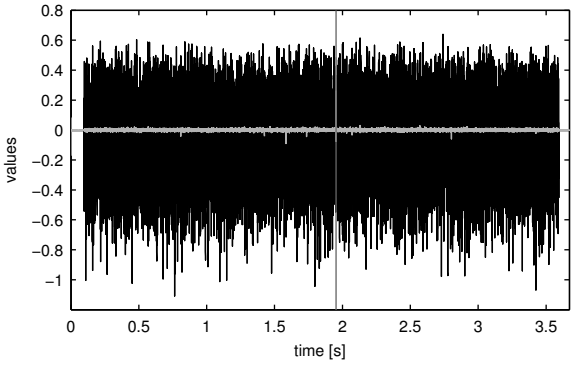


Fig. 2. Performance of the Wiener-Hammerstein model for the benchmark data. Shown are the simulated values in black and the residuals in gray. The vertical gray line separates the estimation set on the left from the validation set on the right.

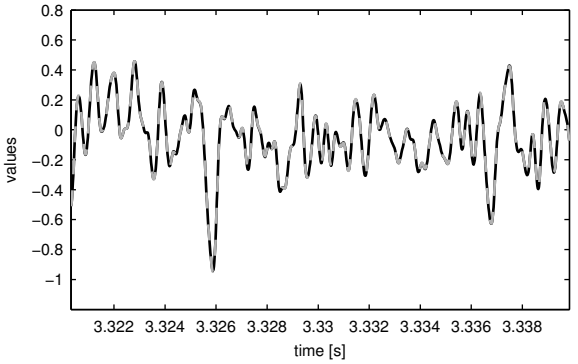


Fig. 3. Performance of the Wiener-Hammerstein model for the benchmark data. Zoom of validation data subsection. The solid black line corresponds to simulated values whereas the dashed gray line refers to the reference data.

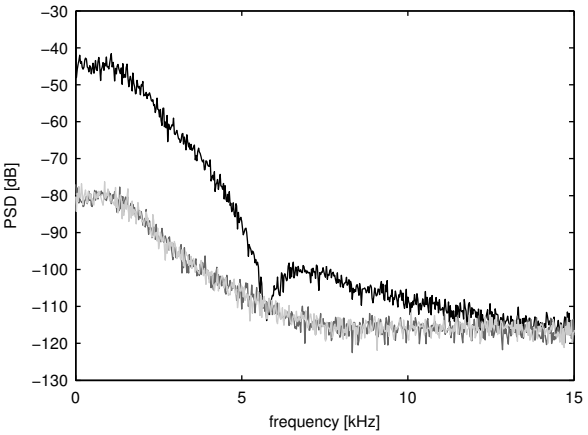


Fig. 4. Power spectral density estimates (using pwelch) of the residuals (estimation: light gray, validation: dark gray) and the true output (black) for the benchmark data.

Table 2. Simulaton performance of different model complexities on the benchmark data. The first value corresponds to the estimation data while the second corresponds to the validation data, both are evaluated for the complete sets. The given values are $10^3 * \text{RMSE}$.

	percentage of full estimation data		
SVs	1	10	20
50	37.6 / 39.5	14.1 / 14.5	13.9 / 13.8
200	30.3 / 32.2	7.4 / 7.8	6.3 / 6.6
500	30.8 / 32.6	6.3 / 6.6	4.3 / 4.5
with regularization on linear parameters $\boldsymbol{c}$			
50	14.1 / 15.0	12.2 / 12.6	12.5 / 13.1
200	27.1 / 28.9	6.5 / 6.8	5.2 / 6.2

data it is not feasible to use all of it for a kernel based method. Thus the fixed size implementation from Section 3 is employed.

The model estimation was carried out it in two phases. In the first phase the model order was selected and the second phase was used to refine the hyperparameters of the model. In both phases the hyperparameters are selected using a grid search and the performance is evaluated using 3-fold cross-validation. For the first phase a model with reduced complexity with 500 support vectors (SVs) is trained on a 20% subset of all training data. The optimal model order was found to be $D = 15$, $R_1 = 12$ and $R_2 = 15$. The final model was obtained in phase two using the whole estimation set and 1000 SVs.

Figure 2 shows the simulation values and the estimation residuals of the final model over time. In Figure 4 the same quantities are depicted but in the frequency domain. Table 1 summarizes the following error measures for the benchmark data: empirical mean $\hat{\mu} = \frac{1}{T'}\sum_{t=1}^{T'} e_t$, empirical standard deviation $\hat{s} = \sqrt{\frac{1}{T'}\sum_{t=1}^{T'}(e_t - \hat{\mu})^2}$, root mean squared error $\text{RMSE} = \sqrt{\frac{1}{T'}\sum_{t=1}^{T'} e_t^2}$ and $\text{fit} = 100\frac{\|\hat{\boldsymbol{y}}-\boldsymbol{y}\|}{\|\boldsymbol{y}-\bar{\boldsymbol{y}}\|}$ with $\bar{\boldsymbol{y}} = \mathbf{1}\mathbf{1}^T\boldsymbol{y}$. All quantities are computed for the training and the evaluation set. To exclude transient effects the first 1000 samples are not included in the computation. An overview of the performance of models with different complexities is given in Table 2. These models only use a fraction of the complete estimation data for training and have been evaluated for different numbers of SVs.

4.3 Regularization

Empirical evidence suggests that regularizing the linear parameters $\boldsymbol{c}$ improves the prediction performance, supporting data is shown in Table 2. The regularization is achieved by adding an additional ridge to the system in (18) and the fixed size expression corresponding to (14). The new regularization $\boldsymbol{R} = \boldsymbol{\Gamma}_I + \boldsymbol{R}_c$ takes place of just $\boldsymbol{\Gamma}_I$. The additional ridge is defined by $\boldsymbol{R}_c = \lambda_c \operatorname{diag}([\mathbf{0}_{(R_1 \cdot M)}, \mathbf{1}_{(R_2)}, 0])$ with regularization parameter λ_c.

The original regularization parameter γ trades off the nonlinear model complexity versus training errors. As the data is subject to very little noise – SNR 70 dB [Schoukens et al., 2008] – overfitting is unlikely. Therfore

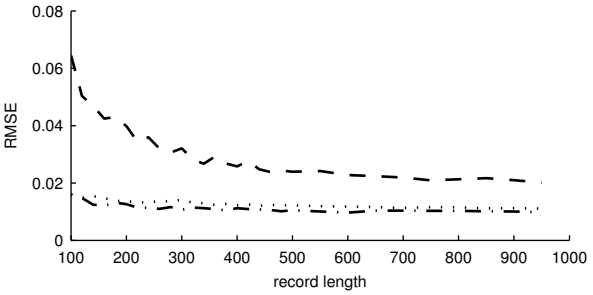
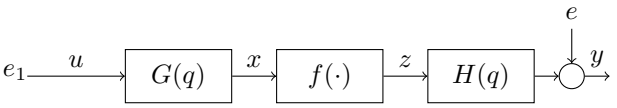


Fig. 5. Prediction performance for several models estimated with different amounts of training data. (dotted line: Wiener-Hammerstein model, dash-dotted line: componentwise model, dashed line: NARX model)

the regularization for the final model of Hammerstein structure (14) can be chosen very small or even dropped completely $\gamma \to \infty$. This corresponds to dropping the matrix $\boldsymbol{\Gamma}_I$. For the overparametrized model in (5) this is not feasible due to the overparametrization.

5. ADVANTAGES OF STRUCTURED MODEL

In this section we will briefly discuss some advantages of the proposed Wiener-Hammerstein model over the fully-black box NARX model. By adding prior knowledge in form of the structure the estimation of a model yielding good prediction performance gets "easier". This should either result in enhanced prediction performance for the model including the structural information or relaxed requirements on the training data. We consider two important scenarios that potentially lead to ill-conditioned estimation problems. On the one hand the perfomance is analysed as a function of the size of the training set, on the other hand the effects of colored inputs during the estimation phase is studied.



As example we will consider a system consisting of the concatenation of $G(q) = q - 0.9$, $f(x) = \operatorname{sinc}(2x)$ and $H(z)$ which has zeros at $z = 1j, 1 \cdot e^{\pm j0.55\pi}$ and poles at $p = 0.48, 0.74 \cdot e^{\pm j0.19\pi}$ (similar to benchmark system with transmission zeros). The remaining experimental settings are a training set with 500 samples, GWN as input and noiseless data. The structured models are trained using the true model orders whereas the NARX model is trained with an increased number of lagged inputs and outputs.

5.1 Results

The amount of input data is critical for every data driven identification approach. Due to the effective parametrization of the Wiener-Hammerstein model, the estimation requires considerably less data than the estimation of the complete black-box model. The behavior of the different models for varying training set sizes is shown in Figure 5.

White spectra are optimal for identification purposes as all frequencies are excited with the same amount of power. In

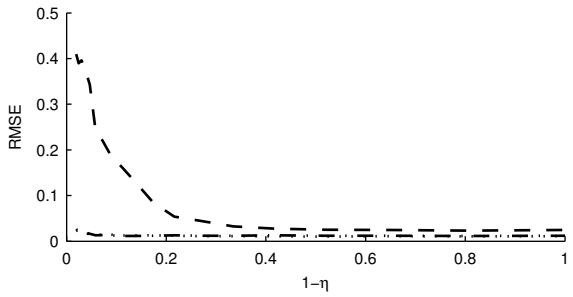


Fig. 6. Prediction performance for models that are trained with colored inputs, the prediction performance is then evaluated for white inputs. (dotted line: Wiener-Hammerstein model, dash-dotted line: componentwise model, dashed line: NARX model)

this section colored excitation signals are investigated. The colored input signal is generated by a low pass IIR filter $u_t = \eta u_{t-1} + n_t$ where n_t is a GWN process and η a control parameter to control the amount of coloring. For values of η close to 1 more and more information is "lost" for high frequencies. This would make the identification of a purely linear systems harder or even impossible. The colored input signal is only used during the estimation phase. For the validation of the approximation performance the system is excited with a white input signal. Thus the models have to generalize to operating regions that were not properly excited during the estimation phase.

All models can effectively handle colored inputs during the training phase even for strongly colored inputs, although the black box model does not reach the same level of accuracy as the other methods. The capability to handle strongly colored inputs is likely due to the nonlinear function that introduces harmonics from the intermediate signal x to the signal z. Therefore the spectrum at the output of the nonlinearity can be better for linear identification than the original input.

6. CONCLUSIONS

This paper studied an approach for identifying a Wiener-Hammerstein system from input/output data only. The approach integrates overparametrization with LS-SVMs, and extends [Goethals et al., 2005]. The key idea is to jointly model the first linear dynamic susbsystem with the static nonlinearity. We then report numerical experiments on how the method behaves under various conditions, and give results on the SYSID2009 benchmark data.

ACKNOWLEDGEMENTS

Research Council KUL: GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), G.0226.06 (cooperative systems and optimization), G.0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite+; Helmholtz: viCERP; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); EU: ERNSI; FP7-HD-MPC (223854); AMINAL. Johan Suykens is a professor and Bart De Moor is a full professor at K.U. Leuven.

REFERENCES

E.-W. Bai. An optimal two stage identification algorithm for Hammerstein-Wiener nonlinear systems. In *Proc. American Control Conference 1998*, volume 5, pages 2756–2760, 1998.

E.-W. Bai and M. Fu. A blind approach to Hammerstein model identification. *IEEE Transactions on Signal Processing*, 50(7):1610–1619, 2002.

N. J. Bershad, P. Celka, and S. McLaughlin. Analysis of stochastic gradient identification of Wiener-Hammerstein systems for nonlinearities with Hermite polynomial expansions. *IEEE Transactions on Signal Processing*, 49(5):1060–1072, 2001.

S. Billings and S. Y. Fakhouri. Identification of a class of nonlinear systems using correlation analysis. *Proceedings of IEE*, 125(7):691–697, 1978.

F. Chang and R. Luus. A noniterative method for identification using Hammerstein model. *IEEE Transactions on Automatic Control*, 16(5):464–468, 1971.

K. De Brabanter, P. Dreesen, P. Karsmakers, K. Pelckmans, J. De Brabanter, J. A. K. Suykens, and B. De Moor. Fixed-Size LS-SVM Applied to the Wiener-Hammerstein Benchmark. In *Proc. IFAC Symposium on System Identification 2008*, 2008.

M. Enqvist and L. Ljung. Linear approximations of nonlinear fir systems for separable input processes. *Automatica*, 41(3):459–473, Mar. 2005.

I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. De Moor. Identification of MIMO Hammerstein models using least squares support vector machines. *Automatica*, 41(7):1263–1272, 2005.

W. Greblicki and M. Pawlak. *Non-Parametric System Identification*. Cambridge University Press, 2008.

K. Narendra and P. Gallman. An iterative method for the identification of nonlinear systems using a Hammerstein model. *IEEE Transactions on Automatic Control*, 11(3): 546–550, 1966.

R. Nowak and B. Van Veen. Random and pseudorandom inputs for Volterra filter identification. *IEEE Transactions on Signal Processing*, 42(8):2124–2135, Aug. 1994.

M. Pawlak. On the series expansion approach to the identification of Hammerstein systems. *IEEE Transactions on Automatic Control*, 36(6):763–767, 1991.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press Cambridge, Mass, 2002.

J. Schoukens, J. A. K. Suykens, and L. Ljung. Wiener-hammerstein benchmark. Technical report, 2008.

T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, 1989.

P. Stoica and T. Söderström. Instrumental-variable methods for identification of Hammerstein systems. *International Journal of Control*, 35(3):459–476, 1982.

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

A. H. Tan and K. Godfrey. Identification of Wiener-Hammerstein models using linear interpolation in the frequency domain (LIFRED). *IEEE Transactions on Instrumentation and Measurement*, 51(3):509–521, 2002.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

C. K. I. Williams and M. Seeger. Using the Nyström Method to Speed Up Kernel Machines. In *Neural Information Processing Systems 13*, pages 682–688, 2001.