Identification of MIMO Hammerstein Models using Least Squares Support Vector Machines

Ivan Goethals, [1] Kristiaan Pelckmans, [2] Johan A.K. Suykens, [3] Bart De Moor [4]

K.U.Leuven, ESAT-SCD-SISTA
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
email: {firstame.lastname}@esat.kuleuven.ac.be

Abstract

This paper studies a method for the identification of Hammerstein models based on Least Squares Support Vector Machines (LS-SVMs). The technique allows for the determination of the memoryless static nonlinearity as well as the estimation of the model parameters of the dynamic ARX part. The SISO as well as the MIMO identification cases are elaborated. The technique can lead to significant improvements with respect to classical methods as illustrated on a number of examples as no stringent assumptions on the nature of the nonlinearity need to be imposed.

Key words: Hammerstein models, ARX, LS-SVM, MIMO systems, kernel methods

1 Introduction

Throughout the last few decades, the field of linear modelling has been explored to the level that most linear identification problems can be solved with fairly standard and well known tools. Extensions to complex nonlinear models are often desirable, though in many situations Hammerstein models may result in good approximations. Hammerstein models are composed of a memoryless static nonlinearity followed by a linear dynamical system. Many techniques have been proposed for the estimation of the nonlinear part in Hammerstein models, which can be classified in parameteric approaches imposing a specific model structure on the solution (see e.g. [15,3,1,12]), and nonparameteric approaches [7,5,11,8,16,23,10] where the enforced structure is minimal. A special case of the parameteric approach is working under the assumption of the global model being almost linear such that the static nonlinearity can be treated as a nonlinear distortion [19].

Approximation of nonlinear functions recently gained interest due to the presence of powerful computers and theoretical breakthroughs as in the area of splines [24], neural networks [2] and regularization networks [17]. Support Vector Machines (SVMs) [22] and related methods (see e.g. [9]) are a powerful methodology for solving problems in nonlinear classification, function estimation and density estimation. SVMs and related kernel methods have been introduced within the context of statistical learning theory and structural risk minimization. In these methods one solves convex optimization problems, typically quadratic programs. Least Squares Support Vector Machines (LS-SVMs) [21,20] are reformulations of standard SVMs, resulting in linear Karush-Kuhn-Tucker systems which can be solved efficiently. LS-SVMs have been proposed as a class of kernel machines with primal-dual formulations to nonlinear regression, kernel FDA, PCA, CCA, PLS, recurrent networks and control [20].

This paper integrates specific requirements for Hammerstein identification in the primal-dual derivations at the core of LS-SVMs, leading to a set of equations at the size of the number of observations. The solutions can be extracted by solving a linear set of equations followed by a singular value decomposition. Results are derived for the single input-single output (SISO) as well as the multiple input-multiple output (MIMO) case. Major advantages of using the LS-SVM framework over the use of classical methods (as e.g. the use of basis expansions) are found in the use of regularization theory as inherited from statistical learning theory [22,18,20], the straightforward derivation of the results from a constrained op-

[1] I. Goethals is a Research Assistant with the Fund for Scientific Research-Flanders (FWO-Vlaanderen).
[2] K. Pelckmans is a Research Assistant with KULeuven.
[3] J. Suykens is an Associate Professor with KULeuven.
[4] B. De Moor is a Full Professor with KULeuven.

timization problem, the ease in which an additive structure can be enforced without restriction on the linear dynamic part and the freedom one gets in modelling the nonlinearity by the design of an appropriate kernel.

The outline of this paper is as follows: in Section 2, some basic aspects of LS-SVMs, applied to static function estimation are reviewed. In Sections 3 and 4 a method for the identification of nonlinear SISO Hammerstein systems is proposed. In Section 5 the method is extended to MIMO Hammerstein systems. Section 6 gives a short overview of existing techniques to identify Hammerstein systems. In Section 7 the method proposed in this paper is tested and compared to existing methods on a number of examples.

As a general rule, lowercase symbols will be used in this paper to denote column vectors. Uppercase symbols are used for matrices. Elements of matrices and vectors are selected using Matlab-notation, e.g. $A(:,i)$ symbolizes the i$^{\text{th}}$ column of A. Estimates for a parameter x will be denoted by $\hat{x}$.

2 Least Squares Support vector Machines for function approximation

Let $\{x_t, y_t\}_{t=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ be the set of input/output training data with input x_t and output y_t. Consider the regression model $y_t = f(x_t) + e_t$ where $x_1, \ldots, x_N$ are deterministic points (fixed design), $f : \mathbb{R}^d \to \mathbb{R}$ is an unknown real-valued smooth function and $e_1, \ldots, e_N$ are uncorrelated random errors with $E[e_t] = 0$, $E\left[e_t^2\right] = \sigma_e^2 < \infty$. In recent years, Least Squares Support Vector Machines (LS-SVMs) have been used for the purpose of estimating the nonlinear f. The following model is assumed:

$$f(x) = w^T \varphi(x) + b,$$

where $\varphi(x) : \mathbb{R}^d \to \mathbb{R}^{n_H}$ denotes the potentially infinite ($n_H = \infty$) dimensional feature map. The regularized least squares cost function is given as

$$\begin{aligned} &\min_{w,b,e} \mathcal{J}(w,e) = \tfrac{1}{2} w^T w + \tfrac{\gamma}{2} \textstyle\sum_{t=1}^n e_t^2, \\ &\text{subject to : } y_t = w^T \varphi(x_t) + b + e_t, t = 1, \ldots, N. \end{aligned}$$

The relative importance between the smoothness of the solution and the data fitting is governed by the scalar $\gamma \in \mathbb{R}_0^+$ referred to as the regularization constant. The optimization performed is known as a form of ridge regression [4]. In order to solve the constrained optimization problem, a Lagrangian is constructed:

$$\mathcal{L}(w,b,e;\alpha) = \mathcal{J}(w,e) - \sum_{t=1}^N \alpha_t \{w^T \varphi(x_t) + b + e_t - y_t\},$$

with α_t the Lagrange multipliers. Taking the conditions for optimality $\frac{\partial \mathcal{L}}{\partial w} = 0$, $\frac{\partial \mathcal{L}}{\partial b} = 0$, $\frac{\partial \mathcal{L}}{\partial e_t} = 0$ and $\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0$ yields the following set of linear equations

$$\left[\begin{array}{c|c} 0 & 1_N{}^T \\ \hline 1_N & \Omega + \gamma^{-1} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \qquad (1)$$

where $y = \begin{bmatrix} y_1 \ \ldots \ y_N \end{bmatrix}^T$, $1_N = \begin{bmatrix} 1 \ \ldots \ 1 \end{bmatrix}^T$, $\alpha = \begin{bmatrix} \alpha_1 \ \ldots \ \alpha_N \end{bmatrix}^T$, $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$ $\forall i,j = 1, \ldots, N$ with application of the so-called kernel trick (positive definite kernel K). For the choice of the kernel $K(\cdot,\cdot)$, see e.g. [18]. Typical examples are the use of a polynomial kernel $K(x_i, x_j) = (\tau + x_i^T x_j)^d$ of degree d or the RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ where σ denotes the bandwidth of the kernel. The resulting LS-SVM model for function estimation can be evaluated in a new point x_* as

$$\hat{f}(x_*;\theta) = \sum_{t=1}^N \alpha_t K(x_*, x_t) + b,$$

where $\theta = (b, \alpha)$ denotes the solution to (1).

3 Identification of Nonlinear ARX Hammerstein models

Hammerstein systems, in their most basic form, consist of a static memoryless nonlinearity, followed by a linear dynamical system, as shown in Figure 1. The aim of Hammerstein modelling is to model the nonlinearity and estimate the model parameters of the linear system from input/output measurements.

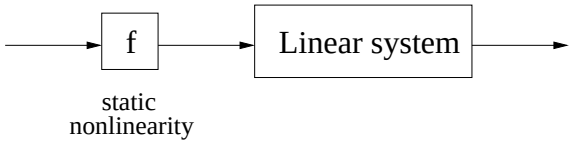

Fig. 1. A Hammerstein system consists of a memoryless static nonlinearity f followed by a linear dynamical system.

In the following derivation, we will restrict ourselves to SISO systems (single input - single output), but as will be shown in Section 5, the presented method is applicable to the MIMO case as well. For the linear dynamical part, we will assume a model structure of the ARX form [13]:

$$y_t = \sum_{i=1}^n a_i y_{t-i} + \sum_{j=0}^m b_j u_{t-j} + e_t, \qquad (2)$$

with $u_t, y_t \in \mathbb{R}, t \in \mathbb{Z}$ and $\{u_t, y_t\}$ a set of input and output measurements. The so-called equation error e_t is assumed to be white and Gaussian, and m and n denote the order of the numerator and denominator in the transfer function of the linear model. The model structure (2)

is generally known as the "AutoRegressive model with eXogeneous inputs" (ARX) and is one of the best known model structures in linear identification. Adding a static nonlinearity $f : \mathbb{R} \to \mathbb{R} : x \to f(x)$ to (2) leads to:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j f\left(u_{t-j}\right) + e_t, \tag{3}$$

which is the general model structure that will be assumed in this paper.

Applying LS-SVM function estimation outlined in the previous section, we assume the following structure for the static nonlinearity f:

$$f(u) = w^T \varphi(u) + d_0.$$

with $\Omega_{ij} = K(u_i, u_j) = \varphi(u_i)^T \varphi(u_j)$ a kernel of choice. Hence, equation (3) can be rewritten as follows:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j \left(w^T \varphi\left(u_{t-j}\right) + d_0\right) + e_t. \tag{4}$$

We focus on finding estimates for the linear parameters $a_i, i = 1, \ldots, n$ and $b_j, j = 0, \ldots, m$ and the static nonlinearity f, parametrized by w and d_0, from a finite set of meaurements $\{u_t, y_t\}$, $t = 1, \ldots, n$. With $r = \max(m, n) + 1$, the resulting optimization function is:

$$\min_{w,a,b,d_0,e} \mathcal{J}(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{t=r}^{T} e_t^2,$$

subject to (4). The Lagrangian of the resulting estimation problem is given by

$$\mathcal{L}(w, d_0, b, e, a; \alpha) = \mathcal{J}(w, e) - \sum_{t=r}^{N} \alpha_t \{ \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j \left(w^T \varphi\left(u_{t-j}\right) + d_0\right)) + e_t - y_t \} \tag{5}$$

The conditions for optimality are given by:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \to w = \sum_{t=r}^{N} \sum_{j=0}^{m} \alpha_t b_j \varphi(u_{t-j}), \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial d_0} = 0 \to \sum_{t=r}^{N} \sum_{j=0}^{m} \alpha_t b_j = 0,$$

$$\frac{\partial \mathcal{L}}{\partial a_i} = 0 \to \sum_{t=r}^{N} \alpha_t y_{t-i} = 0, i = 1, \ldots, n,$$

$$\frac{\partial \mathcal{L}}{\partial b_j} = 0 \to \sum_{t=r}^{N} \alpha_t \left(w^T \varphi(u_{t-j}) + d_0\right) = 0,$$

$$\frac{\partial \mathcal{L}}{\partial e_t} = 0 \to \alpha_t = \gamma e_t, t = r, \ldots, N, \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0 \to (4), t = r, \ldots, N. \tag{8}$$

Substituting (6) and (7) in (8) leads to:

$$\sum_{j=0}^{m} \sum_{q=r}^{N} \sum_{p=0}^{m} b_j \left(b_p \alpha_q \varphi\left(u_{q-p}\right)^T \varphi\left(u_{t-j}\right) + d_0\right) + \sum_{i=1}^{n} a_i y_{t-i} + e_t - y_t = 0, t = r, \ldots, N. \tag{9}$$

If the b_j values were known, the resulting problem would be linear in the unknowns and easy to solve through:

$$\left[\begin{array}{c|c|c} 0 & 0 & \tilde{b} \cdot 1_{N-r+1}^T \\ \hline 0 & 0 & \mathcal{Y}_p \\ \hline \tilde{b} \cdot 1_{N-r+1} & \mathcal{Y}_p^T & \mathcal{K} + \gamma^{-1} I \end{array}\right] \left[\begin{array}{c} d_0 \\ \hline a \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline \mathcal{Y}_f \end{array}\right],$$

with

$$\alpha = \begin{bmatrix} \alpha_r & \ldots & \alpha_N \end{bmatrix}^T, \tilde{b} = \sum_{j=0}^{m} b_j,$$

$$a = \begin{bmatrix} a_1 & \ldots a_n \end{bmatrix}^T, \mathcal{Y}_f = \begin{bmatrix} y_{r+1} & \ldots & y_N \end{bmatrix}^T,$$

$$\mathcal{Y}_p = \begin{bmatrix} y_{r-1} & y_r & \cdots & y_{N-1} \\ y_{r-2} & y_{r-1} & \cdots & y_{N-2} \\ \vdots & \vdots & & \vdots \\ y_{r-n} & y_{r-n+1} & \cdots & y_{N-n} \end{bmatrix},$$

$$\mathcal{K}(p, q) = \sum_{j=0}^{m} \sum_{l=0}^{m} b_j b_l \Omega_{p+r-j-1, q+r-l-1}.$$

Since the b_j are in general not known and the solution to the resulting third order estimation problem (9) is by no means trivial, we will use an approximative method to obtain models of the form (3).

4 An approximative method

4.1 Optimization using collinearity constraints

In order to avoid solving the problem (9), we propose to rewrite (4) as follows:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + d + e_t, \tag{10}$$

which, as will be shown shortly, can conveniently be solved using LS-SVM's. Note, however, that the resulting model class is wider than (4) due to the replacement

of one single w by several vectors $w_j, j = 0, \ldots, m$. The model class (10) is therefore not necessarily limited to the description of Hammerstein systems. A sufficient condition for the estimated model to belong to this class of systems is that the obtained w_j must be collinear in which case w_j is seen as a replacement for $b_j w$. Taking this into account during the estimation leads to extra constraints requiring the angles between any pair $\{w_j, w_k\}, j, k = 0, \ldots, m$ to be zero, or $\left(w_j^T w_k\right)^2 = \sqrt{w_j^T w_j}\sqrt{w_k^T w_k}$. Alternatively, the collinearity constraint can be written as: rank $\begin{bmatrix} w_0 & \ldots & w_m \end{bmatrix} = 1$, which is equivalent to ensuring that a set of $\frac{m(m+1)n_H(n_H-1)}{4}$ 2x2 determinants are zero. As n_H (the dimension of w) is unknown and possibly very high, it is obvious that including such constraints in the Lagrangian would again lead to a non-trivial high-order estimation problem.

Considering that ARX Hammerstein models are contained in the set of models of the form (10), however, we propose to remove the collinearity constraints from the Lagrangian alltogether, solve the more general problem (10), and project the obtained model onto the model-set (4) later. Hereby, we assume that even though collinearity was not explicitly imposed, it will automatically be nearly satisfied in the estimated model of the form (10). Although this approach may seem ad-hoc at first, most existing Hammerstein identification schemes implement similar techniques. In essence, most existing approaches can be summarized as writing the static nonlinearity as a linear combination of general nonlinear basis-functions f_i, each with a certain weight c_i, e.g.

$$f(u_t) = \begin{bmatrix} c_1 & \ldots & c_{n_f} \end{bmatrix} \begin{bmatrix} f_1(u_t) & \ldots & f_{n_f}(u_t) \end{bmatrix}^T,$$

where f_1, f_2, and f_{n_f} are chosen beforehand. This substitution leads to a classical linear identification algorithm where linear model parameters $p_1, p_2, \ldots$ are replaced by vectors $p_1 \begin{bmatrix} c_1 & \ldots & c_{n_f} \end{bmatrix}$. Afterwards, collinearity of these vectors is imposed, e.g. by applying an SVD, and the original model parameters $p_1, p_2, \ldots$ are recovered. Some examples of recently proposed methods employing this approach are found in [16,14].

4.2 Optimization without collinearity constraints

Disregarding the collinearity constraints, the optimization problem that is ultimately solved is the following:

$$\min_{w_j,a,d,e} \mathcal{J}(w_j, e) = \frac{1}{2}\sum_{j=0}^{m} w_j^T w_j + \gamma \frac{1}{2}\sum_{t=r}^{N} e_t^2,$$

subject to

$$\sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + \sum_{i=1}^{n} a_i y_{t-i} + d + e_t - y_t = 0, t = r, \ldots, N, \quad (11)$$

$$\sum_{t=1}^{N} w_j^T \varphi(u_t) = 0, j = 0, \ldots, m, \quad (12)$$

Note the extra constraints (12) to center the nonlinear functions $w_j^T \varphi(\cdot), j = 0, \ldots, m$ around their average over the training set. This to remove the uncertainty resulting from the fact that any set of constants can be added to the terms of the additive nonlinear function (10), as long as the sum of the constants is zero. Removing this uncertainty will facilitate the extraction of the parameters b_j in (3) later. Furthermore, this constraint enables us to give a clear meaning to the bias parameter d, namely $d = \sum_{j=0}^{m} b_j \left(\frac{1}{N}\sum_{k=1}^{N} f(u_k)\right)$.

The resulting Lagrangian is:

$$\mathcal{L}(w_j, d, a, e; \alpha, \beta) = \mathcal{J}(w_j, e) - \sum_{t=r}^{N} \alpha_t \{\sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} w_j^T \varphi(u_{t-j}) + d + e_t - y_t\} - \sum_{j=0}^{m} \beta_j \{\sum_{t=1}^{N} w_j^T \varphi(u_t)\}. \quad (13)$$

Calculating the conditions for optimality, similar as in Section 3, this leads to the following solution:

$$\left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & \mathcal{Y}_p & 0 \\ \hline 1 & \mathcal{Y}_p^T & \mathcal{K} + \gamma^{-1} I & K^0 \\ \hline 0 & 0 & K^{0^T} & \|\Omega\|_F^2 \cdot I_{m+1} \end{array}\right] \left[\begin{array}{c} d \\ \hline a \\ \hline \alpha \\ \hline \beta \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline \mathcal{Y}_f \\ \hline 0 \end{array}\right], \quad (14)$$

where

$$\beta = \begin{bmatrix} \beta_0 & \ldots & \beta_m \end{bmatrix}^T, K^0(p,q) = \sum_{t=1}^{N} \Omega_{t,r+p-q},$$
$$\mathcal{K}(p,q) = \sum_{j=0}^{m} \Omega_{p+r-j-1,q+r-j-1}.$$

4.3 Projecting the unconstrained solution onto the class of NARX Hammerstein models

The projection of the obtained model onto (3) goes as follows. Estimates for the autoregressive parameters $a_i, i = 1, \ldots, n$ are directly obtained from (14). Furthermore,

for the training input sequence $\begin{bmatrix} u_1 & \dots & u_N \end{bmatrix}$, we have:

$$\begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} \hat{\underline{f}}(u_1) \\ \vdots \\ \hat{\underline{f}}(u_N) \end{bmatrix}^T = \begin{bmatrix} \alpha_N & \dots & \alpha_r & & 0 \\ & \alpha_N & \dots & \alpha_r & \\ & & \ddots & & \ddots \\ 0 & & & \alpha_N & \dots & \alpha_r \end{bmatrix} \times \begin{bmatrix} \Omega_{N,1} & \Omega_{N,2} & \dots & \Omega_{N,N} \\ \Omega_{N-1,1} & \Omega_{N-1,2} & \dots & \Omega_{N-1,N} \\ \vdots & \vdots & & \vdots \\ \Omega_{r-m,1} & \Omega_{r-m,2} & \dots & \Omega_{r-m,N} \end{bmatrix} + \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_m \end{bmatrix} \sum_{t=1}^{N} \begin{bmatrix} \Omega_{t,1} \\ \vdots \\ \Omega_{t,N} \end{bmatrix}^T, \tag{15}$$

with $\hat{\underline{f}}(u)$ an estimate for

$$\underline{f}(u) = f(u) - \frac{1}{N} \sum_{t=1}^{N} f(u_t).$$

Hence, estimates for b_j and the static nonlinearity f can be obtained from a rank 1 approximation of the right hand side of (15), for instance using a singular value decomposition. Once all the elements b_j are known, $\sum_{t=1}^{N} f(u_k)$ can be obtained as $\sum_{t=1}^{N} f(u_t) = \frac{Nd}{\sum_{j=0}^{m} b_j}$.

5 Extension to the MIMO case

Technically, an extension of the algorithms presented in the former section to the MIMO case is straightforward, but the calculations involved are quite extensive. Assuming a MIMO Hammerstein system of the form:

$$y_t = \sum_{i=1}^{n} A_i y_{t-i} + \sum_{j=0}^{m} B_j f\left(u_{t-j}\right) + e_t, \tag{16}$$

with $y_t, e_t \in \mathbb{R}^{n_y}$, $u_t \in \mathbb{R}^{n_u}$, $A_i \in \mathbb{R}^{n_y \times n_y}$, $B_j \in \mathbb{R}^{n_y \times n_u}$, $t = 1, \dots, N$, $i = 1, \dots, n$, $j = 0, \dots, m$, and $f : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u} : x \to f(x)$, we have for every row s in (16), that

$$y_t(s) = \sum_{i=1}^{n} A_i(s,:) y_{t-i} + \sum_{j=0}^{m} B_j(s,:) f\left(u_{t-j}\right) + e_t(s). \tag{17}$$

Note that for every non-singular matrix $V \in \mathbb{R}^{n_u \times n_u}$, and for any $j = 0, \dots, m$:

$$B_j(s,:) f\left(u_{t-j}\right) = B_j(s,:) V V^{-1} f\left(u_{t-j}\right). \tag{18}$$

Hence, any model of the form (16) can be replaced with an equivalent model by applying a linear transformation on the components of f and the columns of B_j. This will have to be taken into account when identifying models of the form (16) without any prior knowledge of the nonlinearity involved. In many practical cases, however, the nonlinearity f will be component-wise nonlinear,

$$f(u) = \begin{bmatrix} f_1(u(1)) & \dots & f_{n_u}(u(n_u)) \end{bmatrix}^T, \tag{19}$$

in which case V is automatically restricted to a diagonal matrix, and merely represents the fact that any component of f can be scaled freely, provided that the corresponding columns in the B_j are adapted as well. In the next subsection we will only consider such component-wise nonlinear functions f. Later, general nonlinear functions will be discussed.

5.1 Hammerstein identification with a component-wise nonlinear function

In case f is known to be a component-wise nonlinear function of u, substituting (19) in (17) leads to:

$$\begin{aligned} y_t(s) = & \sum_{i=1}^{n} A_i(s,:) y_{t-i} \\ & + \sum_{j=0}^{m} \sum_{k=1}^{n_u} B_j(s,k) f_k\left(u_{t-j}(k)\right) + e_t(s). \end{aligned} \tag{20}$$

Replacing $B_j(s,k) f_k(u_{t-j}(k))$ by $w_{j,s,k}^T \varphi_k(u_{t-j}(k)) + d_{s,j,k}$ this reduces to

$$\begin{aligned} y_t(s) = & \sum_{i=1}^{n} A_i(s,:) y_{t-i} \\ & + \sum_{j=0}^{m} \sum_{k=1}^{n_u} w_{j,s,k}^T \varphi_k\left(u_{t-j}(k)\right) + d_s + e_t(s), \end{aligned} \tag{21}$$

where

$$d_s = \sum_{j=0}^{m} \sum_{k=1}^{n_u} d_{s,j,k}. \tag{22}$$

Hence, the MIMO Hammerstein identification problem is solved by minimizing the following Lagrangian:

$$\begin{aligned} \mathcal{L}(w_{j,s,k}, d_s, A, e; \alpha, \beta) = & \mathcal{J}(w_{j,s,k}, e) - \\ & \sum_{t=r}^{N} \sum_{s=1}^{n_y} \alpha_{t,s} \{ \sum_{i=1}^{n} A_i(s,:) y_{t-i} + \\ & \sum_{j=0}^{m} \sum_{k=1}^{n_u} w_{j,s,k}^T \varphi_k\left(u_{t-j}(k)\right) + d_s + e_t(s) - y_t(s) \} \\ & - \sum_{j=0}^{m} \sum_{k=1}^{n_u} \sum_{s=1}^{n_y} \beta_{j,s,k} \{ \sum_{t=1}^{N} w_{j,s,k}^T \varphi_k(u_t(k)) \}, \end{aligned}$$

with

$$\mathcal{J}(w_{j,s,k}, e) = \frac{1}{2}\sum_{s=1}^{n_y}(\sum_{j=0}^{m}\sum_{k=1}^{n_u} w_{j,s,k}^T w_{j,s,k} + \sum_{t=r}^{N}\gamma_s e_t(s)^2).$$

The conditions for optimality are:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{j,s,k}} = 0 \rightarrow w_{j,s,k} &= \sum_{t=r}^{N}\alpha_{t,s}\varphi_k(u_{t-j}(k)) \\
&\quad +\beta_{j,s,k}\sum_{t=1}^{N}\varphi_k(u_t(k)), \\
\frac{\partial \mathcal{L}}{\partial A_i(s,:)} = 0 \rightarrow \sum_{t=r}^{N}\alpha_{t,s}y_{t-i} &= 0, \\
\frac{\partial \mathcal{L}}{\partial d_s} = 0 \rightarrow \sum_{t=r}^{N}\alpha_{t,s} &= 0, \\
\frac{\partial \mathcal{L}}{\partial e_t(s)} = 0 \rightarrow \alpha_{t,s} &= \gamma_s e_t(s), \\
\frac{\partial \mathcal{L}}{\partial \alpha_{t,s}} = 0 \rightarrow &\text{(21)}, \\
\frac{\partial \mathcal{L}}{\partial \beta_{j,s,k}} = 0 \rightarrow \sum_{t=1}^{N} w_{j,s,k}^T\varphi_k(u_t(k)) &= 0, \qquad (23)
\end{aligned}$$

with $i = 1,\ldots,n$, $j = 0,\ldots,m$, $k = 1,\ldots,n_u$, $s = 1,\ldots,n_y$, and $t = r,\ldots,N$. The solution is found as:

$$\begin{bmatrix} L_1 & & \\ & \ddots & \\ & & L_{n_y} \end{bmatrix}\begin{bmatrix} X_1 \\ \vdots \\ X_{n_y} \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_{n_y} \end{bmatrix}, \qquad (24)$$

where

$$L_s = \left[\begin{array}{c|c|c|c} 0 & 0 & 1^T & 0 \\ \hline 0 & 0 & \mathcal{Y}_p & 0 \\ \hline 1 & \mathcal{Y}_p^T & \mathcal{K}+\gamma_s^{-1}I & \mathcal{S} \\ \hline 0 & 0 & \mathcal{S}^T & \mathcal{T} \end{array}\right], \quad X_s = \left[\begin{array}{c} d_s \\ \hline \mathcal{A}_s \\ \hline \overline{\alpha}_s \\ \hline \overline{\beta}_s \end{array}\right],$$

$$R_s = \left[\begin{array}{c|c|c|c} 0 & 0 & \mathcal{Y}_{f,s}^T & 0 \end{array}\right]^T,\ \mathcal{Y}_{f,s}^T = \begin{bmatrix} y_r(s)^T & \ldots & y_N(s)^T \end{bmatrix},$$

$$\mathcal{A}_s = \begin{bmatrix} A_1(s,:)^T \\ \vdots \\ A_m(s,:)^T \end{bmatrix}, \overline{\alpha}_s = \begin{bmatrix} \alpha_{r,s} \\ \vdots \\ \alpha_{N,s} \end{bmatrix},$$

$$\overline{\beta}_s = \begin{bmatrix} \beta_{0,s,1} & \ldots & \beta_{m,s,1} & \ldots & \beta_{0,s,d_u} & \ldots & \beta_{m,s,d_u} \end{bmatrix}^T,$$

$$\mathcal{K}(p,q) = \sum_{k=1}^{d_u}\sum_{j=0}^{m}\Omega_{p+r-j-1,q+r-j-1}^k,$$

$$\Omega_{p,q}^k = \varphi_k(u_p(k))^T\varphi_k(u_q(k)),$$

$$\mathcal{T} = \begin{bmatrix} \|\Omega^1\|_F^2 & & 0 \\ & \ddots & \\ 0 & & \|\Omega^{d_u}\|_F^2 \end{bmatrix} \otimes I_{m+1},$$

$$\mathcal{S} = \begin{bmatrix} K^{0,1} & \ldots & K^{0,n_u} \end{bmatrix},$$

$$K^{0,k}(p,q) = \sum_{t=1}^{N}\Omega_{t,r+p-q}^k.$$

Note that the matrices $L_s, s = 1,\ldots,n_y$ are almost identical, except for the different regularization constants γ_s. In many practical cases, however, and if there is no reason to assume that one output is more important than another, it is recommended to set $\gamma_1 = \gamma_2 = \ldots = \gamma_{n_y}$. This will speed up the estimation algorithm since $L_1 = L_2 = \ldots = L_{n_y}$ needs to be calculated only once, but most importantly, it will reduce the number of hyperparameters to tune.

The projection of the obtained model onto (20) is similar as in the SISO case. Estimates for the autoregressive matrices $A_i, i = 1,\ldots,n$ are directly obtained from (24). For the training input sequence $[\, u_1 \ \ldots \ u_N \,]$ and every $k = 1,\ldots,n_u$, we have:

$$\begin{bmatrix} B_0(1,k) \\ \vdots \\ B_m(1,k) \\ \hline \vdots \\ \hline B_0(n_y,k) \\ \vdots \\ B_m(n_y,k) \end{bmatrix}\begin{bmatrix} \underline{\hat{f}_k}^T(u_1(k)) \\ \vdots \\ \underline{\hat{f}_k}^T(u_N(k)) \end{bmatrix}^T = \begin{bmatrix} \beta_{0,1,k} \\ \vdots \\ \beta_{m,1,k} \\ \hline \vdots \\ \hline \beta_{0,n_y,k} \\ \vdots \\ \beta_{m,n_y,k} \end{bmatrix}\sum_{t=1}^{N}\begin{bmatrix} \Omega_{t,1}^k \\ \vdots \\ \Omega_{t,N}^k \end{bmatrix}^T$$

$$+\mathcal{A}\times\begin{bmatrix} \Omega_{N,1}^k & \Omega_{N,2}^k & \ldots & \Omega_{N,N}^k \\ \Omega_{N-1,1}^k & \Omega_{N-1,2}^k & \ldots & \Omega_{N-1,N}^k \\ \vdots & \vdots & & \vdots \\ \Omega_{r-m,1}^k & \Omega_{r-m,2}^k & \ldots & \Omega_{r-m,N}^k \end{bmatrix} \qquad (25)$$

with

$$\mathcal{A} = \left[\begin{array}{cccccc} \alpha_{N,1} & \ldots & \alpha_{r,1} & & & 0 \\ & \alpha_{N,1} & \ldots & \alpha_{r,1} & & \\ & & \ddots & & \ddots & \\ 0 & & & \alpha_{N,1} & \ldots & \alpha_{r,1} \\ \hline & & & \vdots & & \\ \hline \alpha_{N,n_y} & \ldots & \alpha_{r,n_y} & & & 0 \\ & \alpha_{N,n_y} & \ldots & \alpha_{r,n_y} & & \\ & & \ddots & & \ddots & \\ 0 & & & \alpha_{N,n_y} & \ldots & \alpha_{r,n_y} \end{array}\right]$$

and $\underline{\hat{f}_k}(u(k))$ an estimate for

$$\underline{f_k}(u(k)) = f_k(u(k)) - \frac{1}{N}\sum_{t=1}^{N} f_k(u_t(k)).$$

Hence, estimates for the $B_j, j = 0, \ldots, m$, and the static nonlinearities $f_k, k = 1, \ldots, n_u$, can be obtained from the right hand side of (25) using a singular value decomposition. Furthermore, combining (20-22) and (23), we have:

$$\begin{bmatrix} d_1 \\ \vdots \\ d_{n_y} \end{bmatrix} = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=0}^{m} B_j \begin{bmatrix} f_1(u_t(1)) \\ \vdots \\ f_{n_u}(u_t(n_u)) \end{bmatrix},$$

from which estimates for $\frac{1}{N}\sum_{t=1}^{N} f_k(u_t(k))$, $k = 1, \ldots, n_u$, can be obtained. Note that if the row-rank of $\sum_{j=0}^{m} B_j$ is smaller than the column-rank, multiple solutions are possible. This results as an inherent property of blind MIMO Hammerstein identification. In this case, it is up to the user to make a choice out of the possible solutions.

5.2 Hammerstein identification with a general nonlinear function

In case f is a general nonlinear function with

$$f(u) = \begin{bmatrix} f_1(u) & \ldots & f_{n_u}(u) \end{bmatrix}^T, \tag{26}$$

much of the reasoning and notation of the former subsection can be kept. Equation (21) is rewritten as

$$y_t(s) = \sum_{i=1}^{n} A_i(s,:) y_{t-i} + \sum_{j=0}^{m} \sum_{k=1}^{n_u} w_{j,s,k}^T \varphi_k\left(u_{t-j}\right) + d_s + e_t(s),$$

dropping the reference to the k^{th} component of the input vector u_{t-j}. Replacing $\sum_{k=1}^{n_u} \omega_{j,s,k}^T \varphi_k(\cdot)$ by $\omega_{j,s}^T \varphi(\cdot)$ leads to:

$$y_t(s) = \sum_{i=1}^{n} A_i(s,:) y_{t-i} + \sum_{j=0}^{m} \omega_{j,s}^T \varphi\left(u_{t-j}\right) + d_s + e_t(s). \tag{27}$$

The Lagrangian and conditions for optimality follow in the same way as in subsection 5.1. Furthermore, the solution is identical as the one given in subsection 5.1, except for a redefinition of the following parameters:

$$\begin{aligned} \mathcal{K}(p,q) &= \sum_{j=0}^{m} \Omega_{p+r-j-1, q+r-j-1}, \\ \Omega_{p,q} &= \varphi(u_p)^T \varphi(u_q), \\ \mathcal{T} &= \|\Omega\|_F^2 \cdot I_{m+1}, \\ \mathcal{S}(p,q) &= \sum_{t=1}^{N} \Omega_{t, r+p-q}. \end{aligned}$$

The nonlinear function f is obtained as:

$$\begin{bmatrix} B_0(1,:) \\ \vdots \\ B_m(1,:) \\ \hline \vdots \\ \hline B_0(n_y,:) \\ \vdots \\ B_m(n_y,:) \end{bmatrix} \begin{bmatrix} \hat{\underline{f}}^T(u_1) \\ \vdots \\ \hat{\underline{f}}^T(u_N) \end{bmatrix}^T = \begin{bmatrix} \beta_{0,1} \\ \vdots \\ \beta_{m,1} \\ \hline \vdots \\ \hline \beta_{0,n_y} \\ \vdots \\ \beta_{m,n_y} \end{bmatrix} \sum_{t=1}^{N} \begin{bmatrix} \Omega_{t,1} \\ \vdots \\ \Omega_{t,N} \end{bmatrix}^T + \mathcal{A} \times \begin{bmatrix} \Omega_{N,1} & \Omega_{N,2} & \cdots & \Omega_{N,N} \\ \Omega_{N-1,1} & \Omega_{N-1,2} & \cdots & \Omega_{N-1,N} \\ \vdots & \vdots & & \vdots \\ \Omega_{r-m,1} & \Omega_{r-m,2} & \cdots & \Omega_{r-m,N} \end{bmatrix} \tag{28}$$

with $\hat{\underline{f}}(u)$ an estimate for

$$\underline{f}(u) = f(u) - g, \tag{29}$$

and g a constant vector such that:

$$\sum_{j=0}^{m} B_j g = \begin{bmatrix} d_1 \\ \vdots \\ d_{n_y} \end{bmatrix}. \tag{30}$$

Estimates for $\underline{f}$ and the $B_j, j = 0, \ldots, m$, can be obtained through a rank-n_u approximation of the right hand side of (28). If a singular value decomposition is used, the resulting columns of the left hand side matrix of (28) containing the elements of $B_j, j = 0, \ldots, m$, can be made orthonormal, effectively fixing the choice of V in (18).

From $\underline{f}$ in (29) and g in (30), finally, an estimate for the nonlinear function f can be obtained. Note again that if the row-rank of $\sum_{j=0}^{m} B_j$ is smaller than the column-rank, g, and hence the constant terms in f are not uniquely defined. The choice of a particular g is left to the user.

6 Comparison with existing algorithms

As was mentioned in subsection 4.1, most existing Hammerstein identification algorithms can be summarized as writing the static nonlinearity f as a linear combination of general nonlinear basis functions f_k, each with a certain weight c_k,

$$f(u_t) = \sum_{k=1}^{n_f} c_k f_k(u_t), \tag{31}$$

were f_1, f_2, and f_{n_f} are chosen beforehand. Starting from:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j f(u_{t-j}, k),$$

and substituting (31) leads to:

$$y_t = \sum_{i=1}^{n} a_i y_{t-i} + \sum_{k=0}^{n_f} \sum_{j=0}^{m} b_j c_k f_k(u_{t-j}) + e_t \tag{32}$$

$$= \sum_{i=1}^{n} a_i y_{t-i} + \sum_{k=1}^{n_f} \sum_{j=0}^{m} \theta_{j,k} f_k(u_{t-j}) + e_t, \tag{33}$$

which can be solved for $\theta_{j,k} = b_j c_k$, $j = 0, \ldots, m$, $k = 1, \ldots, n_f$ using a least squares algorithm. Estimates for the b_j and c_k are recovered from the SVD of:

$$\begin{bmatrix} \hat{\theta}_{0,1} & \hat{\theta}_{0,2} & \ldots & \hat{\theta}_{0,n_f} \\ \hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \ldots & \hat{\theta}_{1,n_f} \\ \vdots & \vdots & & \vdots \\ \hat{\theta}_{m,1} & \hat{\theta}_{m,2} & \ldots & \hat{\theta}_{m,n_f} \end{bmatrix}. \tag{34}$$

Note that for any set of variables $\epsilon_k, k = 1, \ldots, n_f$ with $\forall u \in \mathbb{R}, \sum_{k=1}^{n_j} \epsilon_k f_k(u) =$ constant and any set $\alpha_j, j = 0, \ldots, m$ such that $\sum_{j=0}^{m} \alpha_j = 0$, $\theta'_{j,k} = \theta_{j,k} + \alpha_j \epsilon_k$ is also a solution to (33), which may lead to conditioning problems and destroy the low-rank property of (34). One possible way to resolve this issue is by calculating:

$$A = \begin{bmatrix} \hat{\theta}_{0,1} & \hat{\theta}_{0,2} & \ldots & \hat{\theta}_{0,n_f} \\ \hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \ldots & \hat{\theta}_{1,n_f} \\ \vdots & \vdots & & \vdots \\ \hat{\theta}_{m,1} & \hat{\theta}_{m,2} & \ldots & \hat{\theta}_{m,n_f} \end{bmatrix} \begin{bmatrix} f_1(u_r) & \ldots & f_1(u_N) \\ f_2(u_r) & \ldots & f_2(u_N) \\ \vdots & & \vdots \\ f_{n_f}(u_r) & \ldots & f_{n_f}(u_N) \end{bmatrix},$$

with $u_k, k = r, \ldots, N$ the inputs of the system, subtracting the mean of every row in A and calculating the SVD of the remaining matrix, from which estimates for the b_j can be extracted. Estimates for the c_k can then be found in a second round by solving (32). Note that this approach amounts to setting the mean of $\hat{f} = \sum_{k=1}^{N} \hat{f}_k$ over the inputs $u_r, \ldots, u_N$ to zero, which is similar to what was done for the LS-SVM, with the exception that in the latter case this constraint was explicitly introduced in the Lagrangian (13).

7 Illustrative examples

7.1 SISO system

The algorithm proposed in this paper was used for identification on the following SISO Hammerstein system:

$$A(z)y = B(z)f(u) + e, \tag{35}$$

with A and B polynomials in the forward shift operator z where $B(z) = z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3$, $A(z) = (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i})(z - 0.97e^{\pm 0.4i})$, and $f : \mathbb{R} \to \mathbb{R} : f(u) = \text{sinc}(u)u^2$ the static nonlinearity.

A Gaussian input sequence u with length 400, zero mean and standard deviation 2 was generated and fed into the system (35). During the simulation the equation noise was chosen white Gaussian with zero mean and as standard deviation 10% of the standard deviation of the sequence $f(u)$. The last 200 datapoints of u and the generated output y were used for identification using the following three techniques:

- **LS-SVM:** The LS-SVM estimation procedure as described in Section 4: The linear system (14) is solved for d, a, α, β. An SVD of the right hand side of (15) is thereafter performed to obtain estimates for the linear system and the static nonlinearity. For the example, an RBF-kernel with $\sigma = 1$ was used. Different values for the regularization parameter γ were tested by applying the obtained model to an independent validation sequence. From these tests $\gamma = 500$ was selected as the best candidate.
- **Hermite:** The general algorithm described in Section 6 with $f_k(u) = e^{u^2}(d^{k-1}/du^{k-1})e^{-u^2}$, the Hermite polynomial of order $k-1$. This expansion was used in [5] for Hammerstein- and [6] for Wiener systems.
- **Gaussian:** The general algorithm described in Section 6 with $f_k(\cdot), k = 1, \ldots, n_f$ localised Gaussian density functions with mean depending on the value of k. As no prior information about the nature of the static nonlinearity is assumed during the identification step, the means of the Gaussian nonlinearities were chosen equidistantly spread between -4 and 4. The variance of the density functions was chosen to be one, in line with the $\sigma = 1$ choice for LS-SVM. The main reason for considering this algorithm is that it is quite closely related to the LS-SVM approach with an RBF-kernel, where the ultimate solution is expressed as a sum of Gaussian density functions around the training datapoints.

100 monte-carlo experiments were performed following the description above with $n = 6, m = 3$. For each experiment and each obtained estimate $\hat{f}$ for the static nonlinearity f, the distance $d = \int_{-4}^{4} \|f(x) - \hat{f}(x)\| dx$ was calculated. The mean and variance of the distances

so obtained using the LS-SVM technique are compared to those obtained from the Hermite- and Gaussian approach using different values for n_f. The results are displayed in Table 1. Note that the LS-SVM technique

Method	mean(d)	std(d)
LS-SVM $\gamma = 500$	0.0064	0.0041
Hermite $n_f = 15$	0.2203	0.7842
Hermite $n_f = 20$	0.7241	2.3065
Hermite $n_f = 25$	1.1217	2.9660
Hermite $n_f = 30$	1.0118	2.9169
Gaussian $n_f = 18$	0.0142	0.0141
Gaussian $n_f = 24$	0.0193	0.1055
Gaussian $n_f = 30$	0.0168	0.0693
Gaussian $n_f = 36$	0.0188	0.0764

Table 1
Mean and variances of obtained distances between estimated and true nonlinearities in a SISO example.

clearly performs much better than the Hermite-approach and about 3 times better than the Gaussian approach. It should be noted at this point that the Hermite-approach was seen to perform better in alternative experiments where a simpler nonlinearity was used, however, for more complex nonlinearities, the Hermite-approach is obviously inferior to the Gaussian and the LS-SVM technique. The two latter ones are similar in nature as in both cases the estimated nonlinearity is written as a sum of Gaussian basis functions with variance 1. However, it should be noted at this point that the RBF-kernel is but one possible choice in the LS-SVM algorithm, and that in principle any positive definite kernel can be chosen. A big disadvantage for the Gaussian approach is that it suffers from overfitting once the parameter n_f is chosen too high, even though with the 200 datapoints available and $n = 6, m = 3$, one could easily go to $n_f = 46$ before the resulting linear equations becomes underdetermined. To avoid this, an extra regularization term $\gamma^{-1} \sum_{k=1}^{n_f} \sum_{j=0}^{n} \theta_{j,k}^2$ can be applied to the estimation problem (33). Results for the Gaussian approach including such a regularization term, and with $n_f = 46$, are displayed in Table 2: Note that the performance of

Method	mean(d)	std(d)
LS-SVM $\gamma = 500$	0.0064	0.0041
Gaussian $\gamma = 10^{13}$	0.0457	0.1028
Gaussian $\gamma = 10^{12}$	0.0089	0.0071
Gaussian $\gamma = 10^{11}$	0.0088	0.0060
Gaussian $\gamma = 10^{10}$	0.0112	0.0086

Table 2
Mean and variances of obtained distances between estimated and true nonlinearities in a SISO example.

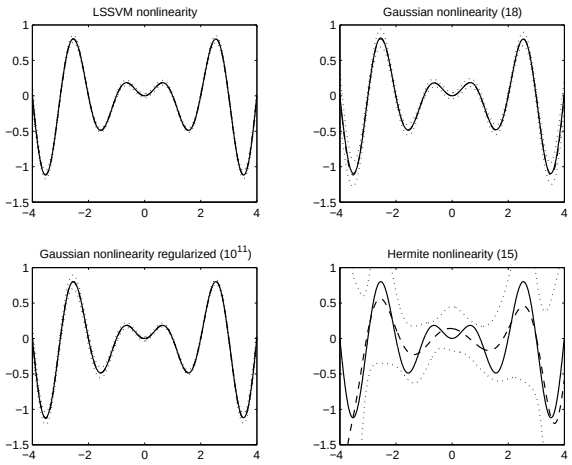

Fig. 2. True nonlinearity (solid) and mean estimated nonlinearity (dashed) for the different techniques compared in a monte-carlo simulation of a SISO system. Results for the LS-SVM algorithm with $\gamma = 500$ are displayed in the top-left figure, those for the Gaussian approach with $n_f = 18$ and without regularization in the top-right figure. The bottom-left figure displays the results for the Gaussian algorithm with $n_f = 46$ and regularization constant $\gamma = 10^1 1$ and the bottom-right figure for the Hermitian algorithm with $n_f = 15$. 90% confidence bounds on the estimated nonlinearities are included in each plot (dotted). The Hermite-approach is obviously inferior to the Gaussian and the LS-SVM technique. The best performance is obtained with the LS-SVM algorithm.

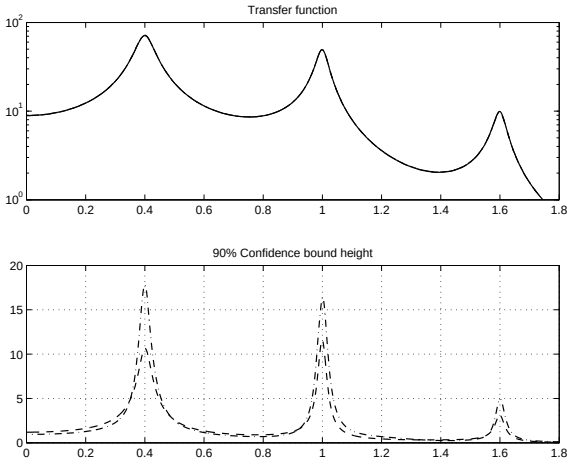

Fig. 3. True transfer function (solid) and mean estimated ones for the LS-SVM estimator (dashed) and the Gaussian estimator with regularization (dash-dotted) in a monte-carlo simulation of a SISO system (top-figure). The width of the 90% confidence bounds on the estimated transfer function (in log norm) is included in the plot below. Note that the transfer functions are visually indistinguishable in the top-figure, but the width of the confidence bounds clearly shows the significantly improved behavior of the LS-SVM approach.

the Gaussian estimator has drastically improved, but is still some 50% worse than the LS-SVM estimator. The same observation can be made by looking at Figure 2, where the true nonlinearity is displayed together with the estimated nonlinearities for the 3 alternative methods described in this section and 90% confidence bounds on these estimates. The estimated linear systems for the LS-SVM and the Gaussian case with regularization ($\gamma = 10^{11}$) are compared in Figure 3. Again, LS-SVM is seen to perform better than the Gaussian approach.

7.2 MIMO system

In a second example, the proposed identification method was applied to the following 2×2 MIMO system:

$$y = \begin{bmatrix} \frac{b_1(z)}{a_1(z)} & \frac{b_2(z)}{a_1(z)} \\ \frac{b_1(z)}{a_2(z)} & \frac{b_2(z)}{a_2(z)} \end{bmatrix} f(u) + e \tag{36}$$

with

$$\begin{aligned} a_1(z) &= (z - 0.98e^{\pm i})(z - 0.98e^{\pm 1.6i}) \\ &\quad (z - 0.97e^{\pm 0.4i}), \\ a_2(z) &= (z - 0.97e^{\pm 0.7i})(z - 0.98e^{\pm 1.4i}) \\ &\quad (z - 0.97e^{\pm 2.3i}), \\ b_1(z) &= z^6 + 0.8z^5 + 0.3z^4 + 0.4z^3, \end{aligned}$$

$$\begin{aligned} b_2(z) &= z^6 + 0.9z^5 + 0.7z^4 + 0.2z^3, \\ f(u) &= \begin{bmatrix} \mathrm{sinc}(u(1)) \\ \sin(u(2)) \end{bmatrix}, \end{aligned}$$

A two-component zero mean white Gaussian input sequence u with length 1000 and standard deviation 2 was generated and fed into the system (36). During the simulation the two components of the equation noise were chosen mutually independent white Gaussian with zero mean and standard deviation 0.1. Based on u and the obtained output y, estimates for $a_1(z), a_2(z), b_1(z), b_2(z)$ and f are obtained using the componentwise MIMO Hammerstein identification algorithm as described in subsection 5.1, whereby $n = 6, m = 3$. As an initial guess, the hyper-parameters were chosen as $\sigma = \gamma_1 = \gamma_2 = 1$. Validation on an independent test set revealed that with these parameters, good results were obtained, so that no further tuning was applied. Results for the obtained nonlinearities are displayed in Figure 4. Estimates for the linear transfer functions appearing in the 2×2 transfer function matrix in (36) are displayed in Figure 5. The resulting fits are clearly quite good, although they could probably still be improved using a proper tuning of the hyper-parameters. On the other hand, this example clearly shows that the LS-SVM method is in fact quite robust against a non-optimal choice of the hyper-parameters. As parameter-tuning is often a time-consuming procedure, this can be seen as a clear advantage of the proposed technique.

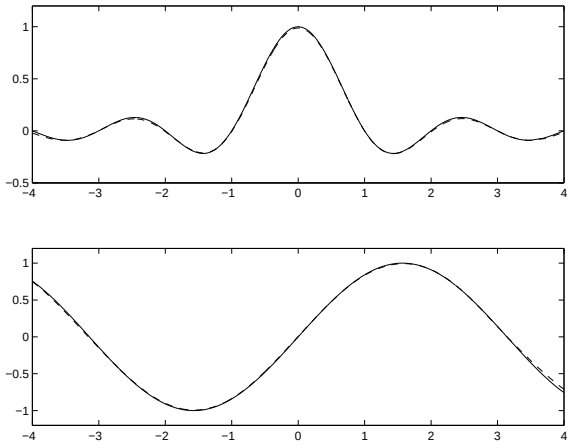

Fig. 4. True nonlinear functions (solid) and estimated ones (dashed) for two components of f in a MIMO example.

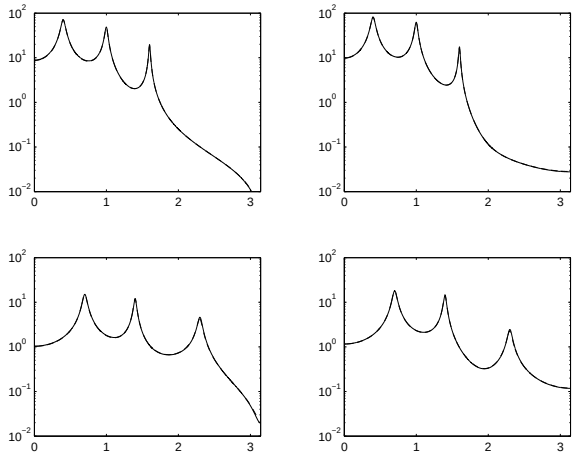

Fig. 5. True (solid) and estimated (dashed) transfer functions in the order as appearing in the Hammerstein system (5), with the top-left figure displaying $b_1(z)/a_1(z)$, the top-right figure displaying $b_2(z)/a_1(z)$, the bottom-left figure displaying $b_1(z)/a_2(z)$, and the bottom-right figure displaying $b_2(z)/a_2(z)$. Note that the true and estimated transfer functions are barely distinguishable.

Note further that in the examples shown, m and n were considered to be known. In practical applications this will often not be the case. As in linear identification problems, several identification runs followed by a selection of the best performing model (e.g. on a validation set) might therefore be necessary to obtain the best possible model. As a general rule, however, a slight overestimation of m and n is, as in the linear case, in general not a problem. In the case of overestimated m for instance certain estimated nonlinearities $w^T\varphi(\cdot)$ will simply be set to zero by the identification algorithm.

8 Conclusions

In this paper, we have proposed a new technique for the identification of MIMO Hammerstein ARX systems. The new method is based on Least Squares Support Vector Machines function-approximation and allows to determine the memoryless static nonlinearity as well as the linear model parameters from a structured linear set of equations. The method was compared to results from two other popular Hammerstein identification algorithms to illustrate its performance. In general the proposed method is seen to work better than these existing methods. This combined with the straightforward derivation of the results, the availablility of a strong regularization framework inherited from statistical learning theory [22,18,20], and the freedom one gets in modelling the nonlinearity by the design of an appropriate positive definite kernel makes the proposed technique an excellent candidate for Hammerstein model identification.

Acknowledgements

Our research is supported by: Research Council KUL: GOA-Mefisto 666, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0240.99 (multilinear algebra), G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0800.01 (collective intelligence), G.0452.04 (new quantum algorithms), G.0499.04 (kernel based methods), research communities (ICCoS, ANMMM); AWI: Bil. Int. Collaboration Hungary/ Poland; IWT: PhD Grants, Soft4s (softsensors), Belgian Federal Science Policy Office: IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling') ; PODO-II (CP/40: TMS and Sustainability); EU: FP5-CAGE; FP5-Quprodis; ERNSI; FP6-BioPattern; Eureka 2419-FliTE; Contract Research/agreements: Data4s, Electrabel, Elia, LMS, IPCOS, VIB;

References

[1] S.A. Billings and S.Y. Fakhouri. Nonlinear system identification using the Hammerstein model. *International Journal of Systems Sciences*, 10:567–578, 1979.

[2] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] F.H.I. Chang and R. Luus. A noniterative method for identification using the Hammerstein model. *IEEE Trans. Automatic Control*, 16:464–468, 1971.

[4] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1989.

[5] W. Greblicki. Non-parametric orthogonal series identification of Hammerstein systems. *International Journal of Systems Science*, 20(12):2355–2367, 1989.

[6] W. Greblicki. Nonparametric identification of Wiener systems by orthogonal series. *IEEE Trans. Automatic Control*, 39(10):2077–2086, 1994.

[7] W. Greblicki and M. Pawlak. Identification of discrete Hammerstein systems using kernel regression estimates. *IEEE, Trans. Automatic Control*, 31:74–77, 1986.

[8] W. Greblicki and M. Pawlak. Nonparametric identification of a cascade nonlinear time series system. *Signal Processing*, 22:61–75, 1991.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, Heidelberg, 2001.

[10] Z. Hasziewicz. Hammerstein system identification by the Haar multiresolution approximation. *International Journal of Adaptive Control and Signal Processing*, 13(8):691–717, 1999.

[11] A. Krzyżak. Identification of discrete Hammerstein systems by the Fourier series regression estimate. *International Journal of Systems Science*, 20:1729–1744, 1989.

[12] Z.H. Lang. Controller design oriented model identification method for Hammerstein systems. *Automatica*, 29:767–771, 1993.

[13] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, NJ, 2 edition, 1999.

[14] T. McKelvey and C. Hanner. On identification of Hammerstein systems using excitation with a finite number of levels. *Proceedings of the* 13^{th} *International Symposiun on System Identification (SYSID2003)*, pages 57–60, 2003.

[15] K.S. Narendra and P.G. Gallman. An iterative method for the identification of nonlinear systems using the Hammerstein model. *IEEE Trans. Automatic Control*, 11:546–550, 1966.

[16] M. Pawlak. On the series expansion approach to the identification of Hammerstein systems. *IEEE Trans. Automatic Control*, 36:736–767, 1991.

[17] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[18] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[19] J. Schoukens, J. G. Nemeth, P. Crama, Y. Rolain, and R. Pintelon. Fast approximate identification of nonlinear systems. *Automatica*, 39:1267–1274, 2003.

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

[21] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[22] V.N. Vapnik. *Statistical Learning Theory*. Wiley and Sons, 1998.

[23] M. Verhaegen and D. Westwick. Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, 63:331–349, 1996.

[24] G. Wahba. *Splines for Observational data*. SIAM, 1990.